

Universidad Católica de Santa María
Facultad de Ciencias e Ingenierías Físicas y Formales
Escuela Profesional de Ingeniería Mecánica,
Mecánica Eléctrica y Mecatrónica



**DISEÑO E IMPLEMENTACIÓN DE UN VEHÍCULO DE
SUPERFICIE NO TRIPULADO PROVISTO DE UN SISTEMA DE
ADQUISICIÓN DE DATOS EN TIEMPO REAL PARA LA
INSPECCIÓN DE CALIDAD DE AGUA Y ANÁLISIS DE
MUESTRAS EN LAGOS**

Tesis presentada por el Bachiller:
Díaz Meza, Rodrigo Alexis

Para optar el Título Profesional de:
Ingeniero Mecatrónico

Asesor de Tesis:
M Sc. Mestas Ramos, Sergio

AREQUIPA – PERÚ

2019



AREQUIPA - PERÚ

Universidad Católica de Santa María

☎ (51 54) 382038 Fax: (51 54) 251213 ✉ ucsm@ucsm.edu.pe 🌐 http://www.ucsm.edu.pe Apartado: 1350

**ESCUELA PROFESIONAL DE INGENIERÍA MECÁNICA, MECÁNICA
ELÉCTRICA Y MECATRÓNICA**

INFORME DICTAMINATORIO

VISTO

EL BORRADOR DE TESIS TITULADO:

**“DISEÑO E IMPLEMENTACION DE UN VEHICULO
DE SUPERFICIE NO TRIPULADO PROVISTO DE UN
SISTEMA DE ADQUISICION DE DATOS EN TIEMPO
REAL PARA LA INSPECCION DE CALIDAD DE
AGUA Y ANALISIS DE MUESTRAS EN LAGOS”**

Presentado por el Bachiller:

DIAZ MEZA RODRIGO ALEXIS

Nuestro **DICTAMEN** es:

Favorable

OBSERVACIONES:

Ninguna

Arequipa, 06 de mayo 2019



ING. SERGIO MESTAS RAMOS



ING. MARCELO QUISPE CCACHUCO

AGRADECIMIENTOS

Quiero expresar mi gratitud a Dios por la vida de mis padres ya que sin ellos nada de esto sería posible, también porque Él ha sido mi guía incondicional durante todo el transcurso de mi vida, quien con sus bendiciones me proporcionó fortaleza para poder afrontar los momentos más difíciles; me dio paciencia para jamás rendirme y sabiduría que permitió culminar con éxito mis metas propuestas hasta ahora.

De igual manera agradezco a mis padres por su amor que me dieron desde mis primeros pasos, su sacrificio, su paciencia y esfuerzo; por haber sido la fuente de motivación para que yo siga adelante y no rendirme nunca, por haberme dado todo y lo mejor de ellos, por educarme, por llenarme de valores, por preocuparse por mí siempre, por estar conmigo mientras veían como tenía noches de estudio o trabajo y por haberme dado ánimos en todo momento a pesar de las adversidades, y es que todo se los debo a ellos y es por ellos que estoy acá. ¡Muchas Gracias Papás!

Doy gracias a todos mis profesores e ingenieros que, desde el primer día de clases en estas aulas, me enseñaron más que asignaturas, me enseñaron a querer mi carrera, y que lo más importante en un profesional es poseer un proyecto de vida. De manera muy especial quiero agradecer a mi tutor de tesis, el ingeniero Sergio Mestas, por haberme no solo guiado y aconsejado en este proyecto sino a lo largo de mi estancia en esta universidad, formándome con ética y valores que deben ir de la mano con mi carrera profesional.

Y estaré eternamente agradecido con cada una de las personas que no he mencionado en este apartado, pero que me ayudaron mucho e hicieron y aportaron para que esto también sea posible.

DEDICATORIA

A:

- Dios por haberme permitido llegar a este punto con salud, con su bendición, con alegría y amor y con todo aquello que me permitió desarrollarme como profesional y hombre de bien.
- Mi madre Gladis Meza, una mujer que me llena de orgullo, la amo, no va a haber manera posible de devolver tanto que me ha ofrecido desde antes que yo estuviera en este mundo. Esta tesis es un logro más que llevo a cabo y es más que seguro que es fruto de tu esfuerzo, dedicación y amor. Este trabajo es por ti y para ti.
- Mi padre Higidio Diaz, quien se esmeró y busco por ofrecerme siempre lo mejor y su amor incondicional, me inculcó muchas enseñanzas y de la que siempre me acordare fue que jamás debo renunciar a pesar de las dificultades y por más que alcance muchos triunfos en la vida siempre la humildad y el respeto de la mano junto con la atención y el servicio al prójimo. Esta tesis también es para ti y te la debo a ti.
- Mis amigos de siempre Daniel, Anthony, Carlos A, Carlos R, Kevin , mis amigos del colegio David, Arturo, Sergio, Renato, Johan, Jesús, Juan, Kevin, Jorge, Gonzalo, a mis amigos de mi facultad y de otras facultades Christian, Marco, José, Paulo, Gustavo, Gerson, Jonathan, Luis, Alejandro, que durante mi niñez, etapa del colegio y estadía en esta gran Universidad he podido formar y compartir no solo ratos de estudio sino también de gratos y divertidos momentos que hicieron que formemos una amistad duradera.

Cabe mencionar que faltan nombres de amigos y de esas personas que son importantes y muy especiales para mí, sin embargo, esto también va dirigido a cada una de ellas porque me ayudaron de una manera desinteresada además que siempre estuvieron conmigo dándome ánimos y apoyándome en todo momento a seguir avanzando y crecer como persona.

CONTENIDO

AGRADECIMIENTOS.....	ii
DEDICATORIA	iii
CONTENIDO.....	iv
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	xi
RESUMEN	xvi
ABSTRACT	xviii
Capítulo I.....	1
1. MARCO METODOLÓGICO	1
1.1 Estado del Arte.....	1
1.2 Descripción del problema	3
1.3 Antecedentes	6
1.3.1 Antecedentes Locales	6
1.3.2 Antecedentes Nacionales	6
1.3.2.1 Vehículos comerciales de superficie no tripulados.....	6
1.3.2.2 Calidad de agua.....	7
1.3.2.3 Trabajos nacionales relacionados al tema propuesto	8
1.3.3 Antecedentes internacionales	10
1.4 Justificación	11
1.4.1 Justificación Tecnológica	11
1.4.2 Justificación Económica	12
1.5 Objetivos.....	12
1.5.1 Objetivo Principal	12

1.5.2 Objetivos Específicos	12
1.6 Metodología	13
1.7 Alcances y limitaciones	13
Capítulo II	14
2. MARCO TEÓRICO	14
2.1 Fuentes y calidad del agua	14
2.2 Esquema general del sistema mecánico y electrónico	14
2.3 Vehículos de superficie.....	15
2.3.1 Submarinos autónomos AUV	15
2.3.2 Operados remotamente ROV	16
2.3.3 De superficie no tripulado USV	17
2.4 Sensores	18
2.4.1 Sensor de temperatura.....	18
2.4.2 Sensor de pH.....	20
2.4.3 Sensor de conductividad eléctrica.....	22
2.4.4 Sensor de oxígeno disuelto	24
2.4.5 Sensor de posicionamiento para navegación	25
2.4.6 Sensor de imagen	27
2.4.7 Sensor para medir la capacidad de líquidos basado en características eléctricas ..	30
2.5 Motores	32
2.5.1 Motores de corriente continua	32
2.5.2 Motores de corriente alterna	33
2.5.3 Servomotores.....	34
2.5.4 Motor a pasos.....	35
2.6 Elementos de control	36

2.6.1 Microprocesadores.....	37
2.6.2 Microcontroladores	39
2.7 Fuentes de Alimentación	42
2.7.1 Batería de plomo ácido	43
2.7.2 Batería de Níquel	44
2.7.3 Batería de litio.....	45
2.8 Medios de comunicación	46
2.8.1 Comunicación alámbrica o cableada	48
2.8.2 Comunicación inalámbrica	51
Capítulo III.....	53
3. DISEÑO DEL SISTEMA	53
3.1 Diseño del vehículo de superficie no tripulado	53
3.1.1 Selección de la forma del vehículo	53
3.1.2 Velocidad del bote	54
3.1.3 Estudios de formas para una embarcación.....	55
3.1.4 Resistencia al avance y velocidad del viento	57
3.1.5 Centro de gravedad	58
3.1.6 Diseño en el software CAD	59
3.1.7 Dibujo en el programa SolidWorks 2017	59
3.2 Esquemas eléctricos del conexionado de sensores y actuadores con los elementos de control.....	65
3.3 Selección de sensores.....	66
3.3.1 Sensor para la medición de pH y temperatura	66
3.3.2 Sensor para la medición de conductividad eléctrica	68
3.3.3 Sensor para la medición de oxígeno disuelto.....	70

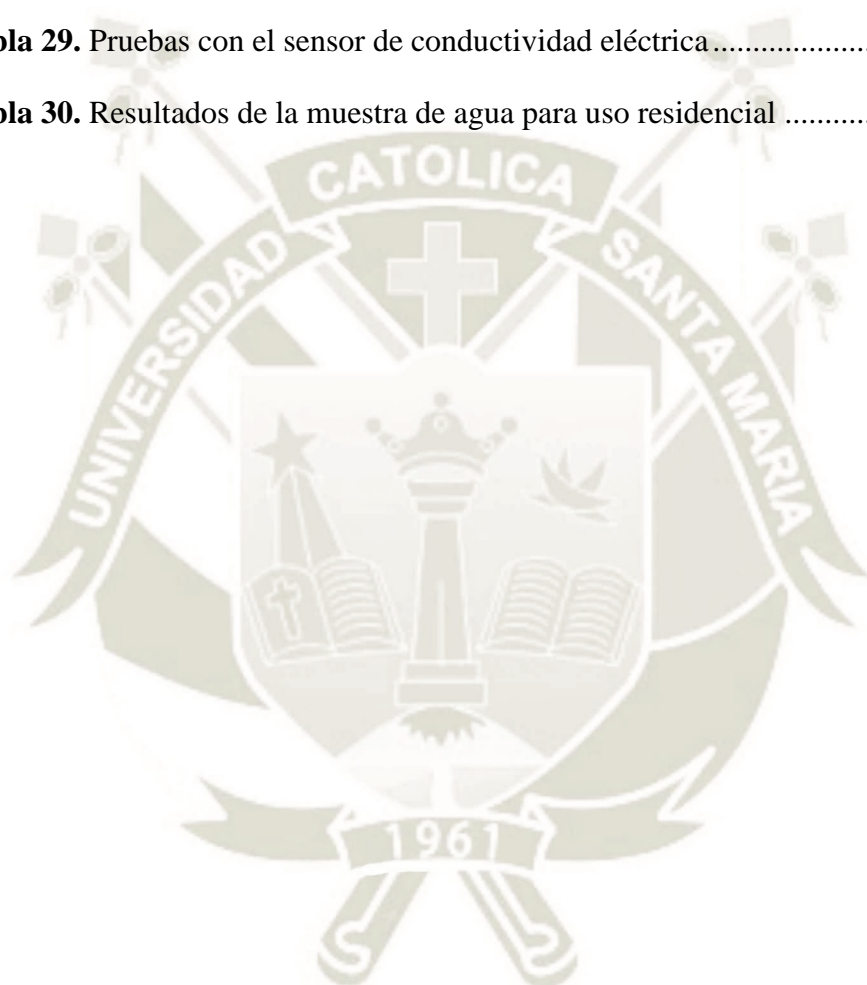
3.3.4 Sensor de posicionamiento	73
3.3.5 Sensor de imagen	74
3.3.6 Sensor para medición de capacidad de líquido	75
3.4 Selección de actuadores	76
3.5 Selección de los elementos de control	77
3.5.1 Selección de plataformas para programación	77
3.5.2 Selección para control de motores	78
3.6 Selección de fuentes de alimentación	79
3.7 Fuente de agua viables	79
Capítulo IV	81
4. SOFTWARE DEL SISTEMA	81
4.1 Dinámica de fluidos computacional CFD	81
4.1.1 Parámetros de mallado	82
4.1.2 Entrada y salida de aire	83
4.1.3 Flujo turbulento	83
4.2 Programa en Arduino nano	84
4.2.1 Diagrama para medición de temperatura	85
4.2.2 Diagrama para medición de pH	86
4.2.3 Diagrama para medición de oxígeno disuelto	87
4.2.4 Diagrama para medición de conductividad eléctrica	88
4.2.5 Diagrama para medición de Flujo	89
4.3 Programa en raspberry pi3 b+	90
4.3.1 Recepción de lectura de sensores en la Raspberry Pi	91
4.3.2 Modelo de datos	92
4.3.3 Programa para activación/desactivación de motores	92

4.3.4 Lectura del GPS.....	94
4.4 Visualización del sistema en Python	94
4.5 Comunicación inalámbrica	95
Capítulo V	97
5. PRUEBAS Y RESULTADOS.....	97
5.1 Velocidad del vehículo de superficie no tripulado	97
5.2 Desplazamiento del vehículo de superficie no tripulado	98
5.3 Pruebas y resultados del análisis CFD.....	100
5.3.1 Velocidad.....	100
5.3.2 Presión.....	102
5.4 Pruebas con los sensores para medición de parámetros de calidad del agua.....	103
5.4.1 Pruebas de pH y temperatura	103
5.4.2 Pruebas de oxígeno disuelto	104
5.4.3 Pruebas de conductividad eléctrica.....	105
5.5 Pantallas de visualización con la aplicación Python.....	107
5.6 Resultados de la muestra de agua para uso residencial	109
5.7 Parámetros de la Calidad del agua medidos en algunos sectores de Arequipa.....	110
5.8 Coordenadas WGS84 tomados de sectores en Arequipa.....	110
CONCLUSIONES	111
RECOMENDACIONES	112
BIBLIOGRAFÍA	113
ANEXOS.....	120

ÍNDICE DE TABLAS

Tabla 1. Disponibilidad del agua al año 2003 de las regiones del Perú.....	2
Tabla 2. Parámetros para la calidad de agua.....	5
Tabla 3. Características de los sensores de temperatura	19
Tabla 4. Conductividad de soluciones	22
Tabla 5. Diferencias de los sensores de imagen	29
Tabla 6. Características de la medición de líquidos.....	31
Tabla 7. Características entre motores unipolares y bipolares.....	36
Tabla 8. Características generales de los elementos de control	37
Tabla 9. Especificaciones de la Raspberry Pi	38
Tabla 10. Especificaciones de los Arduinos	41
Tabla 11. Especificaciones de las baterías	43
Tabla 12. Cuadro comparativo de redes de comunicación	47
Tabla 13. Tipos de cable	48
Tabla 14. Estándares RS-232, RS-422 y RS-485	50
Tabla 15. Medios de comunicación	51
Tabla 16. Velocidad angular y lineal	54
Tabla 17. Régimen de la embarcación	56
Tabla 18. Velocidad del viento en base a la escala Beaufort.....	57
Tabla 19. Características técnicas de la sonda de pH	67
Tabla 20. Conductímetro Marca VA 1022	68
Tabla 21. Modos de operación.....	69
Tabla 22. Características técnicas de la cámara V2.....	75
Tabla 23. Sensores seleccionados	80

Tabla 24. Calidad del agua en algunos sectores de Perú en el año 2016.....	80
Tabla 25. Parámetros de mallado.....	82
Tabla 26. Cálculo de la distancia	99
Tabla 27. Condiciones iniciales de pH y temperatura	103
Tabla 28. Pruebas con el sensor de oxígeno disuelto.....	104
Tabla 29. Pruebas con el sensor de conductividad eléctrica.....	105
Tabla 30. Resultados de la muestra de agua para uso residencial	109



ÍNDICE DE FIGURAS

Figura 1. Contaminación del agua.	3
Figura 2. Vehículos marinos no tripulados.	6
Figura 3. Diagrama esquemático de los sistemas mecánicos, electrónicos y de control.	15
Figura 4. Vehículos de superficie	15
Figura 5. Vehículos submarinos autónomos AUV	16
Figura 6. Vehículos ROV	17
Figura 7. Vehículo de superficie no tripulado	18
Figura 8. Instrumentos de temperatura	19
Figura 9. Medición de voltaje con electrodos de vidrio	21
Figura 10. Medición de voltaje con electrodos ISFET	22
Figura 11. Sensores conductivos	23
Figura 12. Sensores inductivos	24
Figura 13. Medición de oxígeno con sensor amperométrico	25
Figura 14. Medición de oxígeno con sensor óptico	25
Figura 15. Proceso del DME	26
Figura 16. Sistema de navegación por radar	26
Figura 17. Sistema de navegación por GPS	27
Figura 18. Sensor de imagen	27
Figura 19. Sensor CCD	28
Figura 20. Sensor CMOS	29
Figura 21. Medición de nivel de líquidos	30
Figura 22. Funcionamiento del motor de corriente continua	32

Figura 23. Partes del motor de corriente continua	33
Figura 24. Motores de corriente alterna monofásico	33
Figura 25. Motores de corriente alterna trifásico.....	34
Figura 26. Servomotores AC y DC.....	34
Figura 27. Control y giro de un servomotor	35
Figura 28. Motor a pasos	35
Figura 29. Elementos de control	36
Figura 30. Raspberry Pi	37
Figura 31. Bloques internos de un microcontrolador	40
Figura 32. Sistemas Embebidos.....	40
Figura 33. Baterías.....	42
Figura 34. Voltaje vs ciclo de descarga	44
Figura 35. Medios de comunicación.....	47
Figura 36. Conector serial D9.....	49
Figura 37. Estándar RS-422.....	49
Figura 38. Base del vehículo de superficie no tripulado	53
Figura 39. Vehículo de superficie no tripulado	54
Figura 40. Balance de fuerzas sobre una placa plana	55
Figura 41. Centro de gravedad.....	58
Figura 42. Extracción de la carcasa superior	59
Figura 43. Distribución de partes del vehículo	59
Figura 44. Vista superior de la tapa	60
Figura 45. Sólido de la base y tapa del vehículo	60
Figura 46. Medidas iniciales para el cuerpo del tornillo	61
Figura 47. Hélice para el servomotor	61

Figura 48. Ensamble inferior	62
Figura 49. Medidas de la tapa superior delantera	62
Figura 50. Tapa superior delantera	63
Figura 51. Medidas de la tapa superior trasera	63
Figura 52. Tapa superior trasera	64
Figura 53. Vehículo de superficie no tripulado final	64
Figura 54. Circuito electrónico con el Arduino Nano	65
Figura 55. Circuito electrónico con la Raspberry Pi3 B+.....	66
Figura 56. Sonda de electrodo de pH.....	66
Figura 57. Circuito de acondicionamiento para el sensor pH y temperatura.....	67
Figura 58. Medidor de conductividad eléctrica VA 1022	68
Figura 59. Circuito interno del medidor de conductividad eléctrica	69
Figura 60. Diagrama circuital para la medición de conductividad eléctrica	69
Figura 61. Medidor de oxígeno disuelto.....	71
Figura 62. Diagrama interno del medidor para Oxígeno Disuelto	72
Figura 63. Circuito amplificador HA17324A.....	72
Figura 64. Diagrama circuital para la medición de oxígeno disuelto	73
Figura 65. Sensor GPS Neo-6M-0-001	73
Figura 66. Cámara para Raspberry Pi.....	74
Figura 67. Sensor YF-S201	76
Figura 68. Servomotor para el vehículo de superficie no tripulado	77
Figura 69. Mini bomba para succión del líquido	77
Figura 70. Arduino Nano.....	77
Figura 71. Raspberry Pi3	78
Figura 72. Puente H	79

Figura 73. Batería Li-Po y Ni-Cd	79
Figura 74. Modelo en la pantalla Space Claim.....	81
Figura 75. Vehículo junto con el enclosure	82
Figura 76. Entrada y salida de aire	83
Figura 77. Workbench	84
Figura 78. Bosquejo del programa en Arduino	84
Figura 79. Medición de temperatura.....	85
Figura 80. Medición de pH.....	86
Figura 81. Medición de oxígeno disuelto	87
Figura 82. Medición de conductividad eléctrica.....	88
Figura 83. Medición de flujo	89
Figura 84. Estructura de la aplicación	90
Figura 85. Lectura de sensores en la Raspberry Pi3	91
Figura 86. Modelo de datos	92
Figura 87. Estados del sistema.....	92
Figura 88. Toma de muestras.....	93
Figura 89. Lectura del GPS	94
Figura 90. Visualización de la aplicación.....	95
Figura 91. VNC Viewer.....	96
Figura 92. Comunicación inalámbrica entre el Computador y la Raspberry Pi	96
Figura 93. Eje del servomotor	97
Figura 94. Velocidad vista con un tacómetro digital.....	98
Figura 95. Velocidad con dato obtenido del GPS.....	98
Figura 96. Posición WGS-84 obtenido del GPS.....	99
Figura 97. Velocidad	101

Figura 98. Zonas de incidencia con la velocidad.....	101
Figura 99. Presión	102
Figura 100. Zona de incidencia con la presión	102
Figura 101. Condiciones iniciales de pH y temperatura.....	104
Figura 102. Pruebas con el sensor de oxígeno disuelto	105
Figura 103. Pruebas del sensor CE para agua de uso residencial.....	106
Figura 104. Pruebas del sensor CE para agua con azúcar	107
Figura 105. Parámetros de calidad del agua en Python.....	107
Figura 106. Visualización en Google Maps	108
Figura 107. Toma de muestras de líquido	109

RESUMEN

El agua es un recurso de gran importancia para la población en Perú, ya que está destinado al sector agrícola, municipal (generación eléctrica) e industrial. Por tanto, se debe establecer una calidad de agua idónea y verificar ciertos parámetros bajo los estándares de calidad ambiental ECA. Los parámetros por tratar son el pH, temperatura, conductividad eléctrica y oxígeno disuelto.

La presente investigación plantea el diseño e implementación de un vehículo de superficie no tripulado, con la ayuda de una cámara se navega en tiempo real hacia un punto en específico para la toma de muestra del líquido. Luego, se realiza un retorno del vehículo al punto inicial para recoger la muestra y proceder con la medición de los cuatro parámetros de calidad del agua por medio de la plataforma Arduino. Por último, estos datos son mostrados en un sistema de adquisición de datos con la plataforma Raspberry.

El contenido de la tesis consta de cinco capítulos descritos a continuación:

El capítulo I Marco Metodológico presenta una breve descripción de la calidad del agua y del problema para ser analizado. Luego se desarrolla los antecedentes locales, nacionales e internacionales y se presenta una debida justificación tecnológica como económica. Finalmente, se plantean los objetivos, metodología, alcances y limitaciones que tendrá el proyecto.

En el capítulo II Marco Teórico se describe todo el fundamento teórico donde constan los elementos mecánicos, electrónicos y de comunicaciones para que sea de fácil selección al momento de diseñar e implementar un vehículo de superficie no tripulado con el sistema de adquisición de datos.

El capítulo III Diseño del Sistema donde se determina ciertos parámetros como la forma, velocidad, embarcación, resistencia al avance, centro de gravedad y software para el diseño mecánico del vehículo de superficie no tripulado. Además, se realiza el diseño electrónico donde se detallan todos los sensores, actuadores, elementos de control y fuentes de alimentación utilizadas.

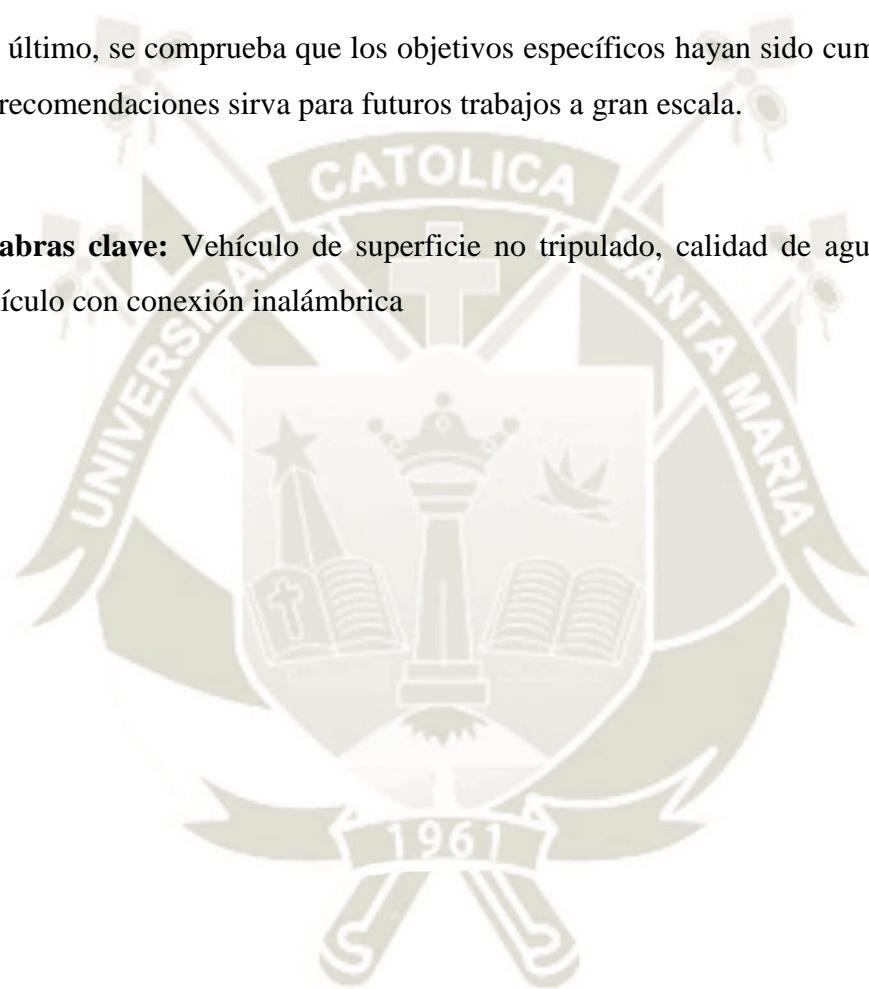
El capítulo IV Software del Sistema consta de la programación realizada, es decir, se utiliza la dinámica de fluidos computacional CFD, programación en las plataformas

Arduino nano y Raspberry Pi3, visualización del sistema desarrollado en Python y comunicación inalámbrica con el sistema mediante el programa VNC Viewer.

El capítulo V Pruebas y Resultados describe las pruebas realizadas para el funcionamiento de cada componente de origen mecánico y electrónico. Además, se realiza un análisis de comparación entre las muestras sacadas del agua con las establecidas mediante la norma ECA.

Por último, se comprueba que los objetivos específicos hayan sido cumplidos, así como las recomendaciones sirva para futuros trabajos a gran escala.

Palabras clave: Vehículo de superficie no tripulado, calidad de agua, VNC Viewer, vehículo con conexión inalámbrica



ABSTRACT

Water is a resource of great importance for the population in Peru, since it is intended for the agricultural, municipal (electric generation) and industrial sectors. Therefore, a suitable water quality should be established and certain parameters verified under ECA environmental quality standards. The parameters to be treated are pH, temperature, electrical conductivity and dissolved oxygen.

The present investigation proposes the design and implementation of a vehicle of unmanned surface, with the help of a camera is navigated in real time to a specific point for the sampling of the liquid. Then, a return of the vehicle to the initial point is made to collect the sample and proceed with the measurement of the four parameters of water quality through the Arduino platform. Finally, these data are displayed in a data acquisition system with the Raspberry platform.

The content of the thesis consists of five chapters described below:

Chapter I Methodological Framework presents a brief description of water quality and the problem to be analyzed. Then the local, national and international background is developed and a due technological and economic justification is presented. Finally, the objectives, methodology, scope and limitations of the project are stated.

Chapter II Theoretical Framework describes the entire theoretical foundation where the mechanical, electronic and communication elements are included so that it is easy to select when designing and implementing an unmanned surface vehicle with the data acquisition system.

Chapter III System Design where certain parameters such as shape, speed, vessel, resistance to advance, center of gravity and software for the mechanical design of the unmanned surface vehicle are determined. In addition, the electronic design is made where all the sensors, actuators, control elements and power supplies used are detailed.

Chapter IV Software of the System consists of the programming carried out, that is to say, the computational fluids dynamics CFD is used, programming in the Arduino nano and Raspberry Pi3 platforms, visualization of the system developed in Python and wireless communication with the system through the VNC program Viewer.

Chapter V Tests and Results describes the tests performed for the operation of each component of mechanical and electronic origin. In addition, a comparison analysis is

carried out between the samples taken from the water and those established by the ECA standard.

Finally, it is verified that the specific objectives have been met, as well as the recommendations for future large-scale work.

Keywords: Unmanned surface vehicle, water quality, VNC Viewer, vehicle with wireless connection



Capítulo I

1. MARCO METODOLÓGICO

1.1 Estado del Arte

El agua es un elemento de gran abundancia vital y se encuentra presente en el planeta, cubre aproximadamente el 70 %, donde existe agua dulce (1%) y salada (69%). El agua dulce por lo general proviene de las cuencas hidrográficas como ríos y arroyos, los cuales se depositan en lagos, lagunas, glaciares, entre otros. La distribución del agua es desigual, no todos los países disponen de este recurso vital (Carrera & Karol, 2001), (Jumapam, 2018).

La extracción del agua a nivel mundial es utilizada para diferentes usos, tales como: agrícolas, municipales (generación eléctrica) e industriales. El nivel de porcentaje de agua utilizados para cada sector es: agrícola 70 %, municipal 11 % e industrial 19 % (FAO, 2016).

- Agropecuario: de acuerdo con la Organización de las Naciones Unidas para la Agricultura y la Alimentación en el año 2011, posee un gran porcentaje del recurso de agua en comparación al resto de sectores. La principal actividad de la agricultura es el riego que depende de varios aspectos como disponibilidad de agua natural, necesidades hídricas para los diferentes cultivos, prácticas y tecnologías utilizadas. El riego es importante para la alimentación mundial.
- Industrial: cada año posee un gran crecimiento y desarrollo económico ya que es utilizado en las centrales termoeléctricas para los procesos de enfriamiento en su mayoría, otros son orientados hacia la industria petrolera, metálica, maderera, procesamiento de alimentos y manufacturera.
- Municipal: juega un papel importante ya que es el fomento de las actividades económicas y mejora de la calidad de vida. Debido a esto, es empleado para el tratamiento de agua, producción de combustibles, generación de vapor y enfriamiento de centrales; en fin, posee grandes impactos potenciales en el uso del agua.

Perú posee recursos naturales ricos referente a la biogenética y ecología, sin embargo, posee problemas con la escasez de agua y tierras agrícolas. Según datos estadísticos, el país posee un 5 % de agua dulce del mundo. La distribución de agua proviene de las

vertientes del Atlántico (97.8 %), Pacífico (1.8 %) y Titicaca (0.5 %). La disponibilidad del agua en las regiones costa, sierra y selva se visualiza en la Tabla 1 (Ercilio, y otros, 2005).

Tabla 1

Disponibilidad del agua al año 2003 de las regiones del Perú

Regional	Disponibilidad	Área	Densidad	Disponibilidad	Usos
	(%)	(Km ²)	(hab/ Km ²)	(millones de m ³)	
Costa	2.0	141,374	100.79	40,870.97	15,557.80
Sierra	18.0	334,156	27.28	367,838.69	3,035.67
Selva	80.0	809,686	4.67	1,634,838.61	379.46
Total	100.0	1,285,216	21.12	2,043,548.26	18,972.92

Fuente: (Ercilio, y otros, 2005).

El agua debe poseer una calidad adecuada y controlada si es destinada para consumo vital. Existen organizaciones como la UNICEF o la Organización Mundial de la Salud que proporciona cierta información estandarizada sobre parámetros de medición y control para la calidad de agua. Las mediciones para el control de calidad son (Buelta & Martínez, 2015):

- Temperatura del grado de calor de agua.
- Conductividad, donde se realizan mediciones de la transmisión de calor o electricidad.
- pH: mide el grado de acidez o alcalinidad de una solución.
- Turbidez producidos por las partículas en suspensión.
- Cloro residual, por lo general usado en la desinfección de las aguas.
- Coliformes: son bacterias con características bioquímicas.

Para realizar un análisis de muestras en lagos y tanques, se procede de la siguiente manera (Buelta & Martínez, 2015):

- El envase debe sumergirse aproximadamente 30 centímetros en un área representativa.
- Evitar áreas de agua tranquilas o estancadas.
- En el caso de que haya corriente, realizar la muestra contra corriente (lagos).

Cada parámetro que corresponde al análisis de calidad de agua puede ser medido mediante instrumentos análogos y digitales. Los instrumentos más conocidos son: pH-metros, fotómetros, medidores de conductividad, termómetros, turbidímetros y equipos químicos de control. En la actualidad, el manejo de los instrumentos de medida se realiza de formar manual a cargo de una persona asignada (PCE, 2014).

Gracias a los avances tecnológicos, se pueden integrar un sistema para realizar inspecciones de calidad de agua desde cierto punto sin la necesidad de estar presente en el sitio.

1.2 Descripción del problema

Como se sabe la contaminación del agua (Figura 1) puede producirse por vertimiento de (Ianver, 2010):



Figura 1. Contaminación del agua

Fuente: (Ocola, 2010)

- Aguas negras o servidas. Esto ocurre en zonas urbanas ya que vierten sus aguas hacia los ríos (Piura, Tumbes, Ucayali, etc.), lagos (Titicaca, Junín) y al mar. Por lo general los desagües contienen excrementos, detergentes, petróleo u otras sustancias tóxicas para las plantas y animales acuáticos. Adicionalmente si no se realiza un tratamiento adecuado provoca enfermedades.

- Los relaves mineros contienen hierro, cobre, zinc, arsénico y más sustancias tóxicas. Estos daños son provocados por los centros mineros y las concentradoras.
- Basuras y desmontes en las orillas del mar, lagos y ríos sin ningún cuidado. Este tipo de vertimiento se mira especialmente cerca de las ciudades e industrias.
- Desechos industriales y químicos tales como: abonos, petróleo, ácidos, etc. Por lo general viene dado de los centros petroleros, industrias de harina, aceite de pescado, textilerías, mineras, etc.

En la región de Arequipa existen cuencas hídricas que abarca aproximadamente el 22 % de agua a nivel del Perú, de los cuales se utiliza una quinta parte debido al mal uso del agua. La contaminación del agua influye en la salud, por lo general en niños y mujeres debido a la falta de hierro y algunas infecciones de patógenos (microbios) (Linares, Tapia, & Chávez, 2017).

Los efectos de contaminación en lagos y embalses son (Sánchez D. , 2016):

- Altas concentraciones de oxígeno disuelto se producen debido a una gran cantidad de agua y la superficie expuesta a la atmósfera.
- Sólidos en suspensión se producen por la acumulación de capas de fangos al fondo del lago.
- Gérmenes patógenos que desaparecen debido a la radiación ultravioleta que ingresa a la superficie de la masa de agua.
- El abundante consumo de dióxido de carbono CO₂, produce la alcalinización del agua.
- Estas grandes cantidades de nutrientes puede provocar un desequilibrio del ecosistema, también llamado eutrofización. Si existe un exceso de nutrientes puede provocar un crecimiento de las algas y un deterioro de la calidad del agua.

En fin, la contaminación puede ocasionar un daño al medio ambiente, ser humano y especies endémicas (aves, peces y anfibios). Existen normas para determinar la calidad de agua en lagos y lagunas que se visualizan en la Tabla 2. Adicional se indican muestreos realizados en la Laguna Moronacocha (promedio de 3 muestras) y en la bahía interior de Puno (promedio de 12 muestras), Lago Titicaca.

Tabla 2.

Parámetros para la calidad de agua

Parámetros	Unidad	Norma ECA	Laguna Moronacocha 2016	Bahía interior de Puno 2010 – 2011
Temperatura	°C	Δ 3	Δ 1.5	Δ 2.8
Conductividad	uS/cm	1000	36.67	1666.95
Oxígeno disuelto (valor mínimo)	mg/l	≥ 5	4.67	6.62
Potencial de hidrógeno	pH	6.5 a 9.0	5.17	9.43

Fuente: (MINAM, 2015), (Minaya, 2017) y (Beltrán, Palomino, Moreno, Peralta, & Montesinos, 2015)

Se pueden realizar mediciones para determinar la calidad de agua de forma manual, pero se corre el riesgo de contaminar el agua y/o afectar la salud de la persona que realiza la toma de muestras. También se recurre a equipos para las mediciones de los parámetros mencionados en la Tabla 2, pero debe disponer los siguientes elementos (Buelta & Martínez, 2015):

- Espacio y mesa disponibles que deben estar desinfectados para el análisis de muestreo.
- Placas de cultivo o Petri.
- Recipientes y estufa para los procesos de esterilización.
- Una nevera para conservar las muestras y cultivos.

El traslado de las muestras tomadas en tiempo real desde algún lago hasta el espacio disponible puede verse afectadas por el mal uso humano. Otro recurso para la toma de muestras es mediante equipos especializados, pero implica un costo relativamente alto.

1.3 Antecedentes

1.3.1 Antecedentes Locales

En las Universidades ubicadas en Arequipa, tales como: Nacional de San Agustín, Católica de Santa María, Católica de San Pablo y la Institución Educativa Emblemática Independencia Americana no poseen algún tema relacionado con el presente proyecto.

1.3.2 Antecedentes Nacionales

1.3.2.1 Vehículos comerciales de superficie no tripulados

Existe una gran variedad de métodos para determinar los parámetros de calidad de agua donde se hace posible que estos estudios se desarrollen mediante la intervención no humana, es decir, por medio de vehículos no tripulados: aéreos (drones) o marítimos (barcos). Para el presente estudio, se establecen vehículos de superficie comerciales vistos en la Figura 2 (Quijarro, 2015):



Figura 2. Vehículos marinos no tripulados

Fuente: (Quijarro, 2015)

- Submarinos autónomos (AUV): la fuente de alimentación está dentro del submarino.
- Operados remotamente (ROV): usado en exploraciones submarinas, es decir, para recolección de muestras y datos submarinos. Su fuente de alimentación es remota (conexión física mediante cables).
- Vehículos de superficie (USV), puede ser de tipo alámbrica o inalámbrica con la estación de control. Por lo general, es utilizado en investigación batimétrica, detección y seguimiento de contaminación, etc.

1.3.2.2 Calidad de agua

Los análisis para muestreo de agua se deben realizar mediante estándares de calidad, uno de ellos es el Estándar de Calidad Ambiental ECA, el cual establece medidas del nivel de concentración de las sustancias presentes en el agua y que no afecte a la salud de las personas y al ambiente. Existen entidades para el cumplimiento de los ECA para agua y son (OEFA, 2015):

- Ministerio del Ambiente, el cual aprueba los estándares de calidad ambiental para agua.
- La Autoridad Nacional del Agua “ANA” ejerce acciones de control, fiscalización y sanción para asegurar la calidad del agua.
- El Organismo de Evaluación y Fiscalización Ambiental “OEFA” fiscaliza el cumplimiento de la ECA.
- Mediante gobiernos regionales y locales junto con la Autoridad Nacional del Agua evitan conflictos de competencia y establecen logros para estandarizar los parámetros de calidad de agua.

Según ANA, los parámetros a determinar la calidad de agua en campo son: temperatura, conductividad, pH, oxígeno disuelto y caudal. Cuando se realiza el trabajo de campo para la inspección del agua, deben utilizarse ciertos materiales, ropa de protección, equipos y personal capacitado, tales como (Fernández, 2011):

- Materiales generales y de laboratorio:
 - Material cartográfico.
 - Fichas de registro y libreta de campo.
 - Etiquetas para la identificación de frascos.
 - Soga y balde plástico transparente.
 - Brazo telescópico muestreador.
 - Papel secante, envases, cintas.
 - Frascos de polietileno, vidrio ámbar, vidrio transparente y estériles.
 - Guantes descartables.
 - Buffers de pH y conductividad.
 - Reactivo para preservación de muestras, goteros y agua destilada.

- Equipos:
 - Sistema de posicionamiento global “GPS”.
 - Multiparámetro.
 - Cámara Fotográfica.
 - Linterna de mano.
 - Correntómetro.
 - Cronómetro.
- Ropa de protección personal:
 - Zapato de seguridad.
 - Pantalón.
 - Chaleco.
 - Casco.
 - Impermeable.
 - Chaleco salvavidas.

Existen sensores que permiten realizar mediciones de los parámetros para la calidad de agua. Pueden basarse por parámetro, es decir, uno a la vez; o tener un sensor que mida todos los parámetros necesarios para la calidad de agua, señalados a continuación:

- Sensor de temperatura para ambiente acuático: lm-35, sonda ds18b20, entre otros (Alldatasheet, 2018).
- Medidores de pH: portátiles, de banco y electrodos (Omega, 2018).
- Sensores comerciales para medir conductividad, de tipo CDH221, PHH424, etc (Omega, 2018).
- Sensores multiparámetros de la marca Eureka. Miden todos los parámetros como: temperatura, pH, conductividad, oxígeno disuelto y profundidad (Eureka, 2017).

1.3.2.3 Trabajos nacionales relacionados al tema propuesto

En el País, se describen tesis relacionados con el tema propuesto:

- **“DISEÑO DE UN USV PARA EL MONITOREO DE CALIDAD DE AGUA Y TOMA DE MUESTRAS EN LAGOS” (Balbuena, 2017)**

El trabajo consiste en realizar diseños mecánicos y electrónicos para la toma de muestras y lograr un monitoreo de la calidad de agua en lagos. Para ello ocupa los siguientes elementos:

- Vehículo de superficie no tripulado, el cual es operado de forma remota.
- El proceso de recolección de muestras se realiza automáticamente con el uso de un cabestrante eléctrico y un sensor de profundidad.
- Realiza un proceso de selección de elementos electrónicos para determinar la localización, direccionamiento, comunicación y alimentación del sistema.

- **“VEHÍCULO AUTÓNOMO DE SUPERFICIE PARA REALIZAR ESTUDIOS BATIMÉTRICOS” (Tejada, 2017)**

Mediante la metodología de diseño VDI2206 se realizan los estudios batimétricos en lagos y lagunas por medio del vehículo de superficie autónomo. Consta de los siguientes elementos para hardware y software:

- Uso de un equipo flotante de tipo catamarán para la colocación de todos los elementos que compone el sistema.
- Utiliza un sensor de posicionamiento global GPS y un magnetómetro para establecer la posición y dirección del vehículo.
- Utiliza un transductor ultrasónico para captar el relieve marino.
- Por último, utiliza un sistema de telemetría para monitoreo y control continuo del vehículo.
- Utiliza varios softwares para ejecutar el sistema, tales como: Arduino IDE, Mission Planer y el Dr. Depth. A fin de tener la ruta trazada y mostrar la profundidad de la toma de muestreos.

- **“DISEÑO DE PLATAFORMA DE ACOPLA Y MINIROV PARA VEHÍCULOS MARINOS DE SUPERFICIE” (Ramírez, 2018)**

El diseño consta de un vehículo marino operado de forma remota, para que realice trabajos de inspección y monitoreo de la calidad del agua. Posee un receptor Wifi para la comunicación inalámbrica de datos. El vehículo miniROV tiene un sistema de

comunicación acústico con el propósito de determinar con precisión la posición debajo del agua.

- **“DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE GESTIÓN DE UNA RED DE SENSORES APLICADA A LA MONITORIZACIÓN DE LA CALIDAD AMBIENTAL DEL RÍO NAPO” (Aragon, 2014)**

Posee un sistema modular compuesto por:

- Un middleware, donde cada usuario sea capaz de compartir y gestionar los datos obtenidos de los sensores.
- Un servicio web y una base de datos para el almacenamiento de la monitorización ambiental.

1.3.3 Antecedentes internacionales

A continuación, se describen tesis de ámbito internacional:

- **“DISEÑO DE UN ALGORITMO DE PATH PLANNING PARA UN DRONE ACUÁTICO DE SUPERFICIE BASADO EN GRAFOS EULERIANOS” (Espartza, 2017)**

El trabajo consiste en realizar un algoritmo de path planning que mida la calidad del agua en diferentes puntos del Lago Ypacaraí. La búsqueda de trayectoria del drone se ha realizado mediante circuitos Hamiltonianos, Algoritmos Genéticos y Eulerianos.

- **“DISEÑO DE MODELOS DE CALIDAD DEL AGUA MEDIANTE EL USO DE LA PERCEPCIÓN REMOTA” (Sepúlveda, 2011)**

Utiliza imágenes satelitales estima algunos parámetros de calidad del agua. Estos modelos son generados mediante el uso de la percepción remota y datos de campo. La percepción remota consiste en leer datos de la superficie de la Tierra y el medio ambiente a cierta distancia mediante sensores ubicados en aviones o plataformas satelitales.

- **“TECNOLOGÍA SIG PARA EL MONITOREO DE LA CALIDAD DEL AGUA EN VERACRUZ” (Zamora, Rustrian, Sánchez, & González, 2017)**

El uso de la herramienta “Sistemas de Información Geográficas SIG”, permite gestionar la información registrada y mostrar de forma gráfica los diversos contaminantes durante un periodo de años. El SIG permite realizar actividades de monitoreo del agua y puede ser utilizado por cualquier organización.

- **“ADAPTACIÓN E IMPLEMENTACIÓN DE UN SISTEMA AUTÓNOMO DE BAJO COSTE DE MONITORIZACIÓN DE CALIDAD DEL AGUA EN TIEMPO REAL” (El-Khoury, 2016)**

Para la monitorización de calidad del agua utiliza componentes de fácil acceso, tales como: placa Arduino Mega, sensores de temperatura, pH, conductividad eléctrica y turbidez con los respectivos circuitos de acondicionamiento. Para la visualización de datos utiliza una pantalla display LCD y guarda los datos en una tarjeta SD.

- **“MEDICIÓN Y CONTROL TELEMÉTRICO DE APROVECHAMIENTOS DE AGUAS NACIONALES Y DESCARGAS RESIDUALES” (Rivera, 2012)**

Este trabajo realiza un estudio de los equipos comerciales disponibles que incluyen: mediciones de los parámetros de calidad y cantidad de agua, seguimiento de la información receptada desde cualquier área, enlace a una página web y transmisión de datos ya sea mediante internet o radiofrecuencia. Utiliza la técnica llamada Biosensor, a fin de mejorar la calidad de aguas residuales.

1.4 Justificación

1.4.1 Justificación Tecnológica

Gracias al avance tecnológico, el desarrollo conjunto de sistemas mecánicos y electrónicos podrán ayudar con la inspección de calidad de agua y evitaría el uso de equipamiento adicional (guantes, lentes, goteros y ropa de trabajo) para toma de muestras en forma manual. Además, evitaría que el ser humano esté expuesto a posibles soluciones peligrosas.

Actualmente, el país cuenta con una variedad de normas ambientales para un mejoramiento en la calidad de las aguas mediante el uso de protocolos de monitoreo, vigilancia, supervisión y fiscalización. Alcanzando estos valores determinados por el

ECA, se evitará que el agua no cause enfermedades, no afecte a la vida acuática ni a la biodiversidad. Por lo tanto, se debe realizar un diseño e implementación de un vehículo de superficie no tripulado a fin de poder sustraer una muestra de agua para determinar mediante sensores los parámetros de calidad de agua; a su vez con el desarrollo de una interfaz de comunicación para la obtención de estos parámetros, con ello se logra obtener mediciones seguras de pH, temperatura, conductividad y oxígeno disuelto e indicar si el agua es o no saludable.

1.4.2 Justificación Económica

El presente trabajo se aplicará para fuentes de agua de portes regulares debido a temas de presupuesto económico del proyecto; teniendo en cuenta que, será proyectado para lagos y dando posibles alternativas a futuro. Por lo tanto, se necesita investigar y desarrollar componentes mecánicos y eléctricos económicos y adaptables entre sí.

1.5 Objetivos

1.5.1 Objetivo Principal

Diseñar e implementar un vehículo de superficie no tripulado provisto de un sistema de adquisición de datos en tiempo real para la inspección de calidad de agua y análisis de muestras en lagos.

1.5.2 Objetivos Específicos

- Diseñar un sistema mecánico (embarcación y mecanismos), un sistema electrónico y de control (hardware y software) que garantice la flotabilidad y estabilidad del sistema.
- Planificar y establecer la trayectoria que desarrollará el vehículo, así como los parámetros para el desplazamiento (velocidad y distancia).
- Establecer los rangos y parámetros para la inspección de la calidad de agua (pH, temperatura, conductividad y oxígeno disuelto).
- Desarrollar un sistema de adquisición de datos para la visualización e inspección de los parámetros realizados por el vehículo.

1.6 Metodología

Se aplicará el método exploratorio debido a que los campos de la mecánica y electrónica son muy versátiles, se investigará los diferentes tipos de tecnologías existentes ya que cada día existe un mejoramiento de ellos. Además, se utilizará el método comparativo ya que se va a analizar y comparar las diferentes alternativas, para escoger la que mejor se adapte a nuestra necesidad logrando así dar una solución al problema.

Por último, se utilizará el método deductivo para exponer las conclusiones obtenidas como resultado de la investigación del proyecto.

1.7 Alcances y limitaciones

- Se establecerá los rangos y parámetros para la inspección de la calidad de agua (pH, temperatura, conductividad y oxígeno disuelto) de los dispositivos electrónicos existentes en el mercado.
- Se investigará y definirá el tipo de fluidos dependiendo del tipo de sensores que puedan ser importados al País.
- Se planificará y establecerá la trayectoria, velocidad y distancia que desarrollará el vehículo utilizando sensores y cámaras.
- La trayectoria del vehículo estará limitada por la capacidad de la fuente de alimentación.
- Se diseñará e implementará un sistema mecánico y electrónico para la medición y comunicación de parámetros de calidad del agua garantizando estabilidad y flotabilidad.
- Se desarrollará un sistema de adquisición de datos para la visualización e inspección de los parámetros realizados por el vehículo.
- Se diseñará e implementará un tanque de medio metro donde se podrán realizar las pruebas de los sistemas mecánicos y electrónicos.

Capítulo II

2. MARCO TEÓRICO

Ante de comenzar con la descripción del sistema mecánico y electrónico, se va a explicar sobre los tipos de agua que existen en Perú.

2.1 Fuentes y calidad del agua

Como se ha mencionó en el apartado 1.1, el agua es un recurso hídrico fundamental y utilizado para diferentes usos. Las fuentes de agua se clasifican de tres maneras: continentales superficiales, aguas marinas y descarga de aguas residuales (Fernández, 2011).

- La calidad del agua en áreas continentales puede variar dependiendo de su entorno físico, químico y antropológico los cuales alteran las características de los recursos hídricos. Por lo habitual, los puntos de monitoreo de calidad del agua se evalúan en ríos, quebradas, lagos y lagunas. La toma de muestras de agua debe ser recogida lo más cerca posible al centro de la fuente de agua.
- Generalmente, las aguas marinas contienen cloruro de sodio o sal, entre otros compuestos disueltos. Por tanto, la ubicación para el monitoreo del agua depende de las fuentes contaminantes y evaluarse para una actividad específica, tales como: embarque de mineral, exploración del petróleo, actividades pesqueras, etc. Las muestras de agua se deben hacer mediante el uso de botellas Niskin.
- Las aguas residuales pueden ser de origen doméstico (aguas usadas) o industrial proveniente de residuos químicos. Por tal motivo, el monitoreo de calidad del agua debe ser analizado antes de que la descarga ingrese a una fuente de agua. Las muestras de agua se deben hacer mediante un recipiente con el debido cuidado.

2.2 Esquema general del sistema mecánico y electrónico

El presente capítulo describe el sistema mecánico (vehículo de superficie no tripulado y sus componentes) y sistema electrónico (sensores, motores, circuitos de control y visualización). Para que un vehículo de superficie no tripulado tenga un control autónomo, se implementa ciertos componentes (Figura 3) que ayudan al vehículo a moverse sin la necesidad de que ser humano esté presente en la embarcación y

dependiendo de los requerimientos para la toma de muestras y mediciones de los parámetros de calidad del agua.

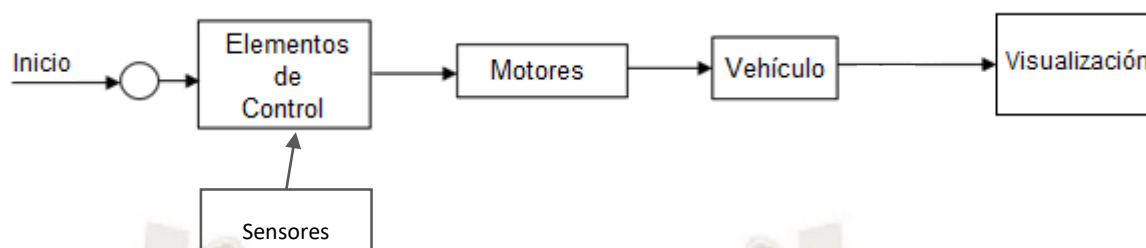


Figura 3. Diagrama esquemático de los sistemas mecánicos, electrónicos y de control

Fuente: Elaboración Propia

2.3 Vehículos de superficie

Los vehículos de superficie son categorizados en tripulados y no tripulados, tal como se visualiza en la Figura 4. De este último, tal como se indicó en el apartado 1.4.2.1, los vehículos de superficie no tripulados se clasifican en submarinos autónomos AUV, operados remotamente ROV y de superficie USV (Quijarro, 2015), (Innova, 2010).

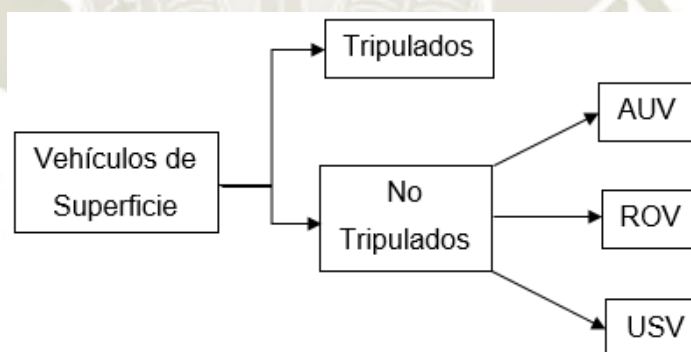


Figura 4. Vehículos de superficie

Fuente: Elaboración Propia

2.3.1 Submarinos autónomos AUV

Este tipo de vehículos son utilizados para la investigación, monitoreo y defensa. Posee las siguientes características (Innova, 2010):

- Dispone de una ruta programada.
- Su desplazamiento es mediante motores o cambios en la densidad del agua.

- Puede realizar muestreos de forma autónoma.

Existen varios tipos de submarinos autónomos, entre los cuales se tiene (Innova, 2010):

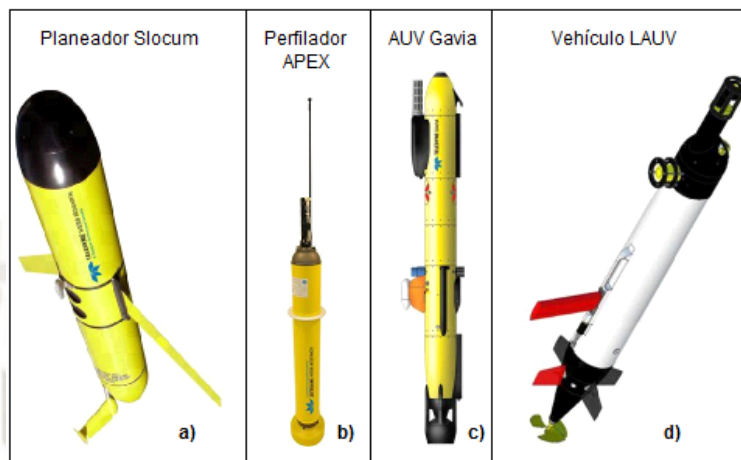


Figura 5. Vehículos submarinos autónomos AUV

Fuente: (Innova, 2010)

- Planeador Slocum, visto en la Figura 5a. Realiza navegaciones de forma autónoma sin consumo apreciable de energía. Gestiona una actividad de toma de datos, comunicaciones, navegación y flotabilidad.
- Perfilador autónomo APEX, visto en la Figura 5b. Es un estilo de boya utilizado para medir variables oceanográficas desde 2000 metros en vertical. Tiene la capacidad de enviar datos vía satélite ARGOS o IRIDIUM.
- AUV Gavia, visto en la Figura 5c. Puede integrarse sensores de otros fabricantes y módulos personalizados de carga útil.
- Vehículo submarino ligero y autónomo LAUV, visto en la Figura 5d. Suele utilizarse en operaciones puntuales como: estudios hidrográficos, operaciones de seguridad y vigilancia o en campañas oceanográficas. Es una herramienta para estudio científico y monitoreo operativo.

Por lo general, estos vehículos suelen equiparse con sensores de turbiedad, fluorescencia, oxígeno disuelto, conductividad eléctrica, entre otros (Innova, 2010).

2.3.2 Operados remotamente ROV

Son vehículos que están conectados mediante cables, posee un dispositivo de control el cual permite maniobrar el vehículo y obtener datos de acuerdo con el equipo instalado.

Dispone de utilizar potencias mayores a las de un vehículo autónomo ya que pueden ser alimentados desde tierra. Su desventaja es que tiene un alcance limitado, por tal razón es empleado en inspecciones de corto alcance. Posee dos tipos de vehículos ROV (Innova, 2010):

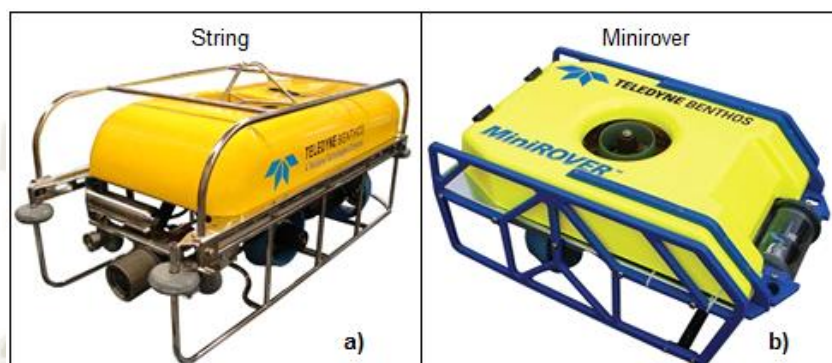


Figura 6. Vehículos ROV

Fuente: (Innova, 2010)

- Stingray, visto en la Figura 6.a. De peso ligero para que cualquier persona pueda manejarlo desde cualquier área estable.
- Minirover, visto en la Figura 6.b. Es más ligero que el anterior prototipo, compacto y portátil. Tiene multiconectividad debido a la variedad de sensores que pueden ser instalados en el vehículo.

2.3.3 De superficie no tripulado USV

En la Figura 7 se dispone de un vehículo de superficie no tripulado, este se moviliza ya sea de forma autónoma sobre el agua sin la necesidad de algún tipo de tripulación. Posee las siguientes ventajas y aplicaciones (Zima, 2018), (Embention, 2018):

- El vehículo puede ingresar a zonas de alto riesgo, donde el ser humano no tiene fácil acceso.
- Son resistentes, estables, rápidos (dependiendo de la carga) y maniobrables.
- Puede ser guiado por GPS (sensor de posicionamiento global).
- Ofrece mayor eficiencia debido al sistema autónomo, eso quiere decir que puede aumentar el tiempo de permanencia en alguna zona acuática.
- Se utiliza en aplicaciones como: batimetrías, exploración, vigilancia y reconocimiento, toma de muestras de calidad y contaminación del agua, etc.



Figura 7. Vehículo de superficie no tripulado

Fuente: (Zima, 2018)

2.4 Sensores

Un sensor es un dispositivo capaz de leer variables físicas como presión, temperatura, humedad, pH, entre otras; y convertirlas en señales compatibles con el sistema de control (Maraña, 2005). Las mediciones por tratarse para la calidad del agua son: temperatura, potencial de hidrógeno pH, conductividad eléctrica y oxígeno disuelto.

2.4.1 Sensor de temperatura

La temperatura es una de las medidas más comunes en cualquier proceso industrial y ambiental. Existen varios tipos de sensores que dependen del tipo de aplicación, ya que algunos fenómenos son influidos por la temperatura, tales como (Creus, 2005):

- Variación en los estados sólidos, líquidos y gaseosos de los cuerpos.
- Variación de resistencia de un conductor o semiconductor.
- Fuerza electromotriz creada a partir de dos metales diferentes, llamados termocuplas.
- Intensidad de la radiación en un cuerpo, etc.

De este modo, en la Figura 8 se indican instrumentos de temperatura: termómetros, termorresistencias (RTD), termocuplas, pirómetros de radiación y sensores de circuito integrado (LM35, DS18B20, etc.). En la Tabla 3, se describen las siguientes características entre los sensores mencionados (Maraña, 2005), (Creus, 2005).



Figura 8. Instrumentos de temperatura
Fuente: (Disetec, 2018), (Instructables, 2018)

Tabla 3.
Características de los sensores de temperatura

Características	RTD	Termocupla	Circuito Integrado
Rango [°C]	-200 a 950 °C	-190 a 1480 °C	0 – 150 °C
Costo	Alto	Bajo	Bajo
Respuesta	Lenta	Rápida	Rápida
Tamaño	Grande	Pequeño	Pequeño
Salida	Casi lineal (Ecuación 1)	No lineal	Lineal (Ecuación 2)
Tipo de salida	Análogo	Análogo	Análogo Digital
Ventajas	Exactitud, estabilidad	Rango de medida	Fácil interfaz

Desventajas	Errores por autocalentamiento	Requiere circuitos de compensación	Rango de temperatura
-------------	-------------------------------	------------------------------------	----------------------

Fuente: (Maraña, 2005), (Creus, 2005)

Ecuación del sensor RTD

$$R = R_o(1 + \alpha * \Delta T) \quad (1)$$

Donde:

- R representa la variación de resistencia a calcular.
- R_o es la resistencia en función de la temperatura de referencia T_o .
- α es el coeficiente de temperatura del conductor a cero grados centígrados.
- ΔT es la diferencia entre la temperatura final y de referencia ($T - T_o$).

Ecuación del sensor LM35

$$V = 10 \left[\frac{mV}{^{\circ}C} \right] * T[^{\circ}C] \quad (2)$$

Donde:

- V es el voltaje de salida del sensor.
- $10 \text{ mV}/^{\circ}C$ es el factor de escala del sensor.
- T es la temperatura por medir.

2.4.2 Sensor de pH

El potencial de hidrógeno sirve para medir la alcalinidad o acidez del agua. Las medidas de pH se encuentran en el rango de 0 a 14. Se dice que el agua pura posee un pH igual a 7, debido a que tiene una concentración de ion hidrógeno de 10^{-7} moles por litro, visto en la Ecuación 3. Cuando el $pH < 7$, se dice que posee una disolución ácida; y si el $pH > 7$, se obtiene una disolución básica (Creus, 2005).

$$pH = \log \frac{1}{[H^+]} \quad (3)$$

Donde el agua pura se calcula de la siguiente manera:

$$\text{pH} = \log \frac{1}{[10^{-7}]} = 7$$

Los métodos comunes para la medición del pH son el sistema de:

- Electrodo de vidrio: posee una membrana de vidrio sensible a los iones hidrógeno de pH. En la parte interna posee un hilo de plata recubierto de cloruro de plata. Cuando se introduce el electrodo en el líquido existe un potencial que está en proporción directa con la concentración del ion hidrógeno del líquido y cambia con la temperatura. Para la medida del potencial, se coloca un segundo electrodo (de referencia) con el propósito de cerrar el circuito y llevar a cabo una diferencia de potencial o voltaje, visto en la Figura 9 (Creus, 2005).

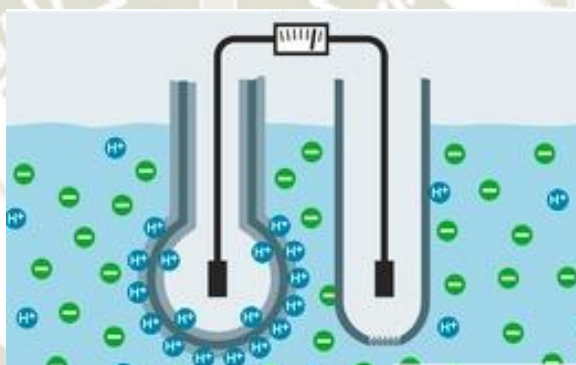


Figura 9. Medición de voltaje con electrodos de vidrio

Fuente: (Endress+Hauser, 2018)

- En la Figura 10 se indica un transistor ISFET (transistor efecto de campo – ion sensitivo). Posee una fuente y las configuraciones drenador D – fuente S del elemento semiconductor. Sobre el transistor existe una acumulación de iones hidrógeno, por medio de la cual la corriente circula entre la fuente y el drenador. Además, posee las siguientes características: (Endress+Hauser, 2018)
 - Respuesta rápida.
 - Funciona en líquidos sucios e impurezas.
 - Posee un electrodo de referencia recambiable.
 - Señal de pH de baja impedancia.
 - Utilizar cuando el producto posea altas cantidades de disolventes orgánicos.
 - Salida de respuesta rápida y
 - Prácticamente irrompible.

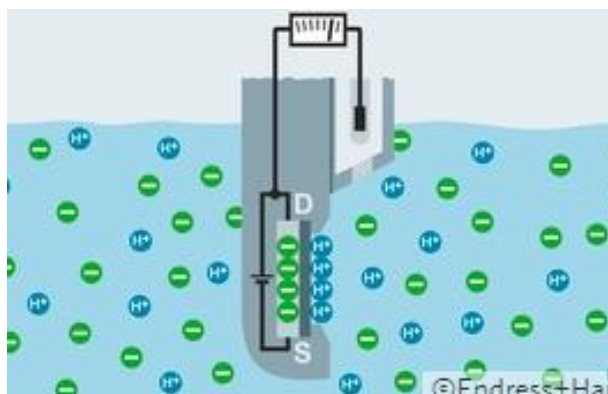


Figura 10. Medición de voltaje con electrodos ISFET

Fuente: (Endress+Hauser, 2018)

2.4.3 Sensor de conductividad eléctrica

La conductividad se produce cuando una corriente eléctrica circula mediante una solución acuosa. Como se sabe, el agua destilada pura no posee corriente, pero gracias a los minerales disueltos, se genera una conducción eléctrica. Esta solución contiene iones positivos y negativos (Creus, 2005).

Tabla 4.

Conductividad de soluciones

Soluciones a 25 [°C]	Conductividad [uS/cm]
Agua ultra pura	0.01 a 0.1
Agua potable	50 a 100
Agua de mar	10000 a 100000
(5, 50) % NaOH Sosa Cáustica	(223000, 150000)

Fuente: (Creus, 2005)

La corriente que existe en la solución y de un circuito externo depende del número y tipo de iones presentes, área efectiva de los electrodos, diferencia de potencial, distancia entre los electrodos y temperatura. La conductividad eléctrica específica se mide como el recíproco de la resistencia, es decir, valor inverso de la resistencia llamado conductancia. En la Tabla 4, se tiene los valores de conductividad de ciertas soluciones (Creus, 2005).

Existen dos tipos de sensores para la conductividad eléctrica: conductivos e inductivos. Dependiendo de la aplicación y rango de medición se escoge el sensor, mostrado a continuación (Endress+Hauser, 2018):

- El sensor conductivo se utiliza cuando se requiere medir conductividades bajas en agua pura y ultra pura. En la Figura 11 se visualizan dos electrodos separados a una distancia y conectados a una fuente de alimentación donde se genera una corriente eléctrica. Los aniones y cationes libres son directamente proporcionales a la conductividad eléctrica. Cuando se requiere ampliar el rango de medición, se dispone de cuatro electrodos.

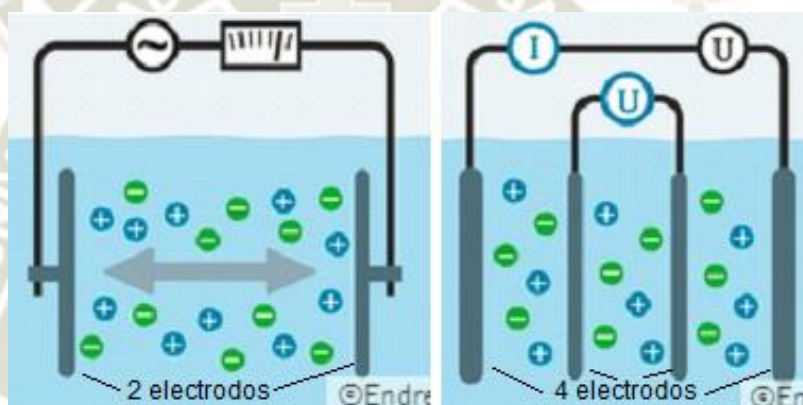


Figura 11. Sensores conductivos
Fuente: (Endress+Hauser, 2018)

- El sensor inductivo se utiliza para medir altas conductividades, por ejemplo: leche, bases, ácidos, etc. En la Figura 12 se visualizan dos bobinas de transmisión y recepción. Los cationes y aniones se mueven cuando se genera un campo magnético debido a la fuente de alimentación conectada, luego se genera una corriente eléctrica en la bobina de recepción. La intensidad o corriente es directamente proporcional al número de iones libres.

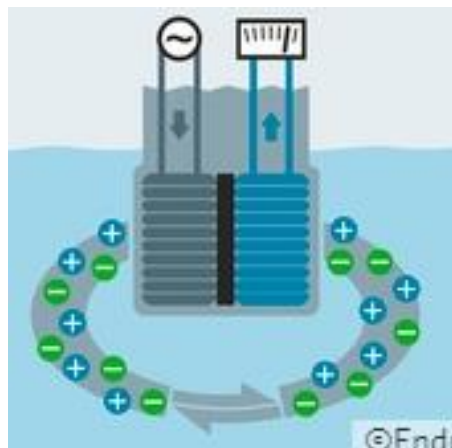


Figura 12. Sensores inductivos

Fuente: (Endress+Hauser, 2018)

2.4.4 Sensor de oxígeno disuelto

Es el oxígeno libre en el agua, quiere decir que no se encuentra combinado con el hidrógeno ni con otros sólidos presentes en el agua. La medición de oxígeno disuelto es de gran importancia para la vida marina. Por lo general, se utiliza en el tratamiento de aguas y control de aireación. Se mide en miligramos de oxígeno por litro de agua en ppm (partes por millón), visto en la Ecuación 4 (Creus, 2005). Donde la densidad del agua es igual 1 g/cm^3 .

$$\text{ppm} = \frac{\text{mg}}{\text{litro}} * \frac{1}{\text{densidad agua} \left(\frac{\text{g}}{\text{cm}^3} \right)} = \frac{\text{mg}}{\text{litro}} \quad (4)$$

Existen dos tipos de sensores para la medición del oxígeno disuelto y son (Endress+Hauser, 2018):

- Los sensores amperométricos poseen buena precisión, linealidad y un amplio rango de medida en la concentración de oxígeno. En la Figura 13 se visualiza como el ánodo y cátodo son sumergidos en un líquido electrolítico donde se aplica un voltaje. El oxígeno atraviesa del ánodo al cátodo generando una corriente eléctrica, ésta es directamente proporcional a la presión parcial de oxígeno.

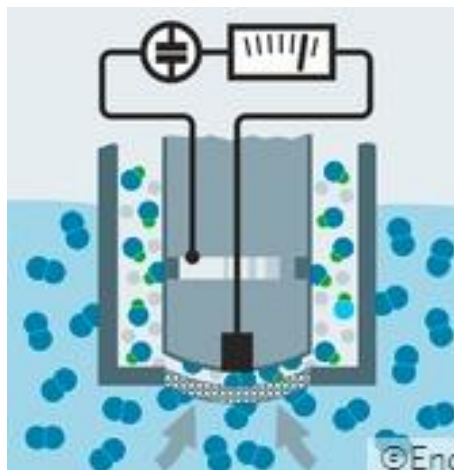


Figura 13. Medición de oxígeno con sensor amperométrico

Fuente: (Endress+Hauser, 2018)

- Los sensores ópticos (Figura 14) poseen una rápida respuesta, mediciones estables y bajo mantenimiento. El sensor está compuesto de un led y un fotodiodo que detecta la intensidad de luz el cual refleja la presión parcial de oxígeno.

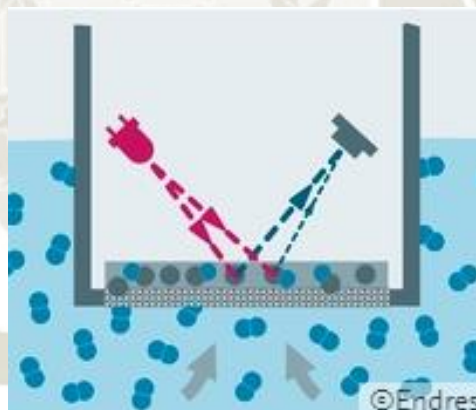


Figura 14. Medición de oxígeno con sensor óptico

Fuente: (Endress+Hauser, 2018)

2.4.5 Sensor de posicionamiento para navegación

La finalidad del sensor de posicionamiento utilizado para navegación es determinar la localización geográfica de un cuerpo. Existen diferentes sistemas de navegación, tales como (López, 2014), (Eltrinauta, 2016):

- Equipos de medida de distancia DME. Está compuesto por un emisor/receptor y el transponder ubicado en tierra. La distancia estimada (Figura 15) se calcula mediante la Ecuación 5 (López, 2014):

$$d = \frac{c \cdot (t - 50 \mu s)}{2} \quad (5)$$

Donde “d” representa la distancia estimada, “c” representa la velocidad de la luz y “t” indica el tiempo que se demora en enviar y recibir la secuencia específica de pulsos.

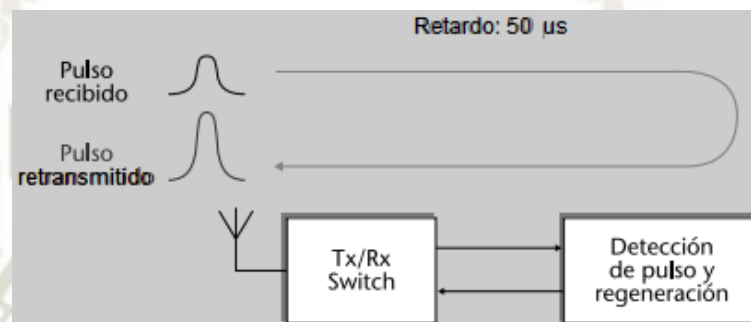


Figura 15. Proceso del DME

Fuente: (López, 2014)

- Radar, visto en la Figura 16. Es un aparato emisor/receptor de ondas electromagnéticas, donde la energía transmitida es interceptada por el cuerpo y reirradiada en todas las direcciones, al final esa señal es enviada a una pantalla donde se recibe la información de la ubicación del cuerpo (Eltrinauta, 2016).

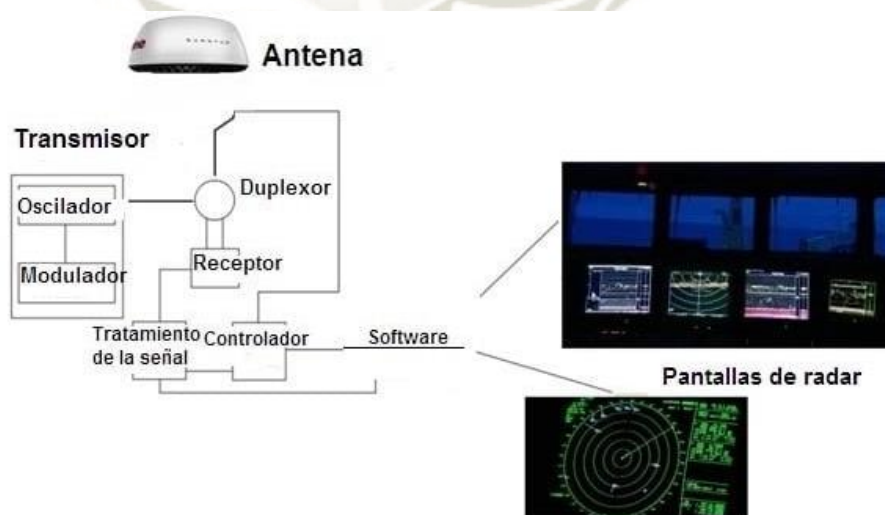


Figura 16. Sistema de navegación por radar

Fuente: (Eltrinauta, 2016)

- Sistema de posicionamiento global GPS, visto en la Figura 17. Es un sistema que brinda posiciones en tres dimensiones, a cada hora, velocidad y capacidad a un número ilimitado de personas. Su función es medir la distancia entre el receptor y algunos satélites (Eltrinauta, 2016).



Figura 17. Sistema de navegación por GPS

Fuente: (Navegar, 2014)

2.4.6 Sensor de imagen

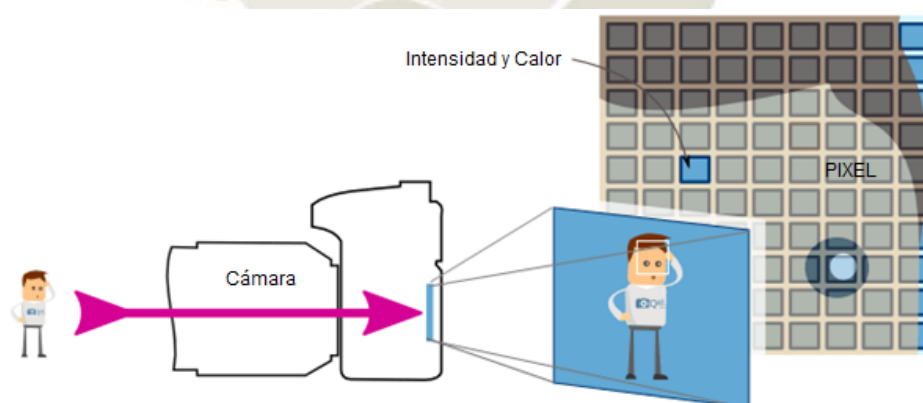


Figura 18. Sensor de imagen

Fuente: (Quecamaraflex, 2018)

El sensor de una cámara está formado por bastantes megapíxeles o celdas fotosensibles a nivel microscópico. Cuando se toma una foto o se realiza un video, cada píxel del sensor

recoge información de luz (intensidad y color), visto en la Figura 18. Cada celda contiene un fotodiodo y un circuito electrónico capaz de convertir la luz en electricidad. El sensor de imagen está compuesto por dos elementos: células fotosensibles y el dispositivo de transferencia de carga (Quecamaraflex, 2018).

Existen dos tipos de sensores: CCD y CMOS.

- Dispositivo acoplado de carga CCD, visto en la Figura 19. Se tiene una imagen, cuando la lectura o carga se transfiere hacia un colector y a su vez se traslada al conversor. Como se visualiza en la Figura, el conversor pone la carga en señal analógica de voltaje, para luego digitalizarla a través de una circuitería auxiliar (Decamaras, 2014).

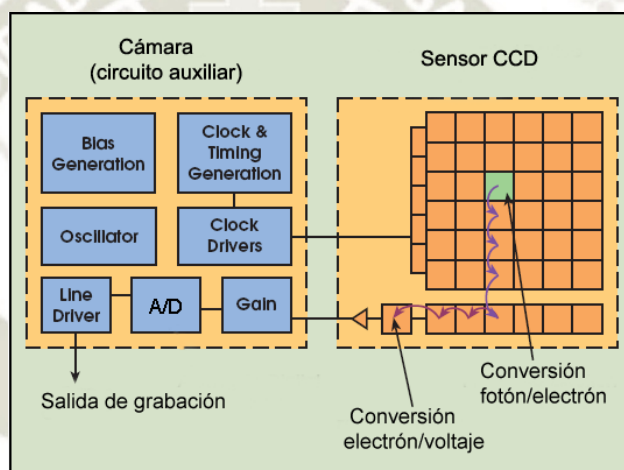
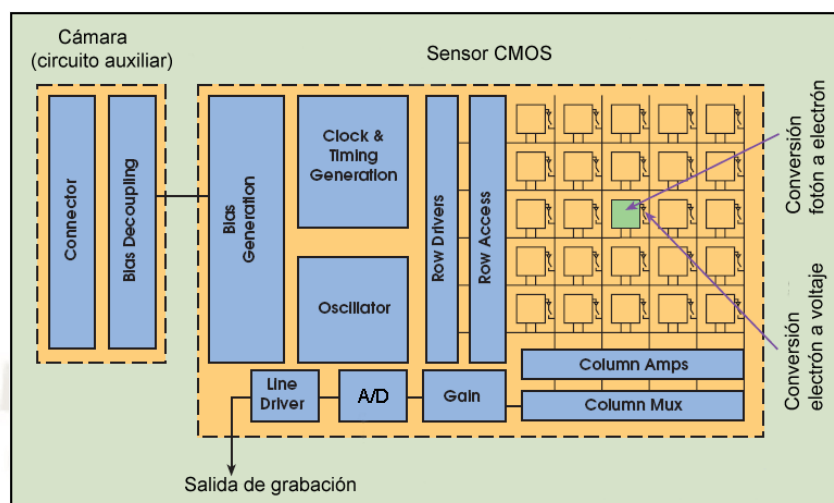


Figura 19. Sensor CCD

Fuente: (Decamaras, 2014)

- Sensor semiconductor de óxido de metal complementario CMOS, visto en la Figura 20. A diferencia del CCD, este sensor conserva la carga y luego la traslada directamente al conversor. Como se visualiza en la Figura, la digitalización se realiza píxel a píxel, por lo tanto, su carga llega a la circuitería auxiliar ya digitalizada (Decamaras, 2014).


Figura 20. Sensor CMOS
Fuente: (Decamaras, 2014)
Tabla 5.

Diferencias de los sensores de imagen

Característica	Sensor	Descripción
Responsividad	CCD < CMOS	Nivel de señal por unidad de energía óptica incidente.
Rango dinámico	CCD > CMOS	Es la relación entre el nivel de saturación de píxeles y el umbral de la señal no captada
Ruido	CCD > CMOS	Interferencia electromagnética de señales
Respuesta Uniforme	CCD > CMOS	Se da cuando un píxel no presenta cambios debido a la incidencia de luz
Velocidad	CCD < CMOS	Nivel de rapidez del proceso de imagen capturada
Enventanado	CCD < CMOS	Es la selección de proporciones de la imagen

Blooming	CCD > CMOS	Se produce cuando un píxel saturado contagia a los demás causando efectos indeseados
----------	------------	--

Fuente: (Decamaras, 2014)

En la Tabla 5, se describen las características que poseen los sensores CCD y CMOS. La elección de cualquier sensor depende de las necesidades, por ejemplo: se utiliza el sensor CCD cuando se desea una alta calidad de imagen, aunque eso represente tener un gran tamaño de la cámara y un consumo elevado. Se escoge el sensor CMOS cuando se requiera una buena velocidad y bajo consumo (Decamaras, 2014).

2.4.7 Sensor para medir la capacidad de líquidos basado en características eléctricas

Existen varios tipos de instrumentos para la medición directa del líquido y se clasifican en resistivo, conductivo, capacitivo, ultrasónico, radiación, láser, entre otros, visto en la Figura 21 (Creus, 2005).

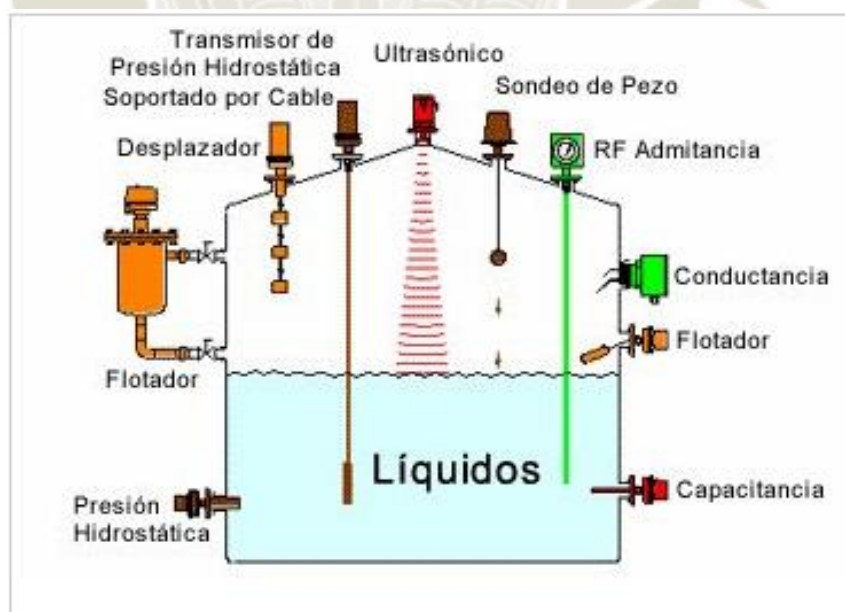


Figura 21. Medición de nivel de líquidos

Fuente: (Antech, 2011)

Tabla 6.

Características de la medición de líquidos

Medidor	Campo de medida	Precisión	Ventaja	Desventajas
Resistivo	Ilimitado	-	Versátil	Líquido conductor
Capacitivo	0,6 m	$\pm 1\%$.	Resistencia, corrosión	Recubrimiento para el electrodo
Ultrasónico	30 m	$\pm 1 - 3\%$.	Todo tipo de tanques y líquidos con espuma y sin contacto con el líquido	Sensible a la densidad
Radiación	2,5 m	$\pm 0.5 - 2\%$.		Fuente radiactiva
Láser	2 m	$\pm 0.5 - 2\%$.		Funciona en ambientes cubiertos
Efecto Hall	-	$\pm 1\%$.	Versátil	Contacto directo al agua

Fuente: (Creus, 2005)

De la Tabla 6 se destaca los siguientes funcionamientos por cada sensor (Creus, 2005):

- El medidor resistivo funciona cuando el líquido es conductivo para excitar el circuito electrónico formado por electrodos y un relé.
- La medición del capacitor se realiza cuando se sumerge el electrodo al líquido y utiliza un transductor para la conversión capacitancia / nivel.
- El sensor ultrasónico consiste en la emisión y recepción de ondas a cierta frecuencia.
- El medidor de radiación emite rayos gamma y debe estar colocado en forma vertical para su medición.
- La medición láser se realiza a través de la reflexión en un espejo sobre una superficie.
- El medidor de efecto Hall consiste en la medición de campos magnéticos.

2.5 Motores

Los motores eléctricos son máquinas cuya función es realizar una cantidad de trabajo, por ejemplo: movimiento de vehículos, accionamiento de puertas eléctricas, traslado de objetos mediante bandas transportadores, entre otras aplicaciones, es decir, un motor convierte la energía eléctrica en mecánica. Dependiendo de los requerimientos, existen diferentes tipos como: servomotores, motor a pasos, motores de corriente alterna y continua (Espinosa, 2016).

2.5.1 Motores de corriente continua

En la Figura 22 se describe el funcionamiento de un motor de corriente continua. Cuando circula la corriente a través de una espira produce una fuerza de tal modo que la espira gire. También se visualiza que la corriente eléctrica circula en un solo sentido gracias al colector de delgas (Chapman, 2012).

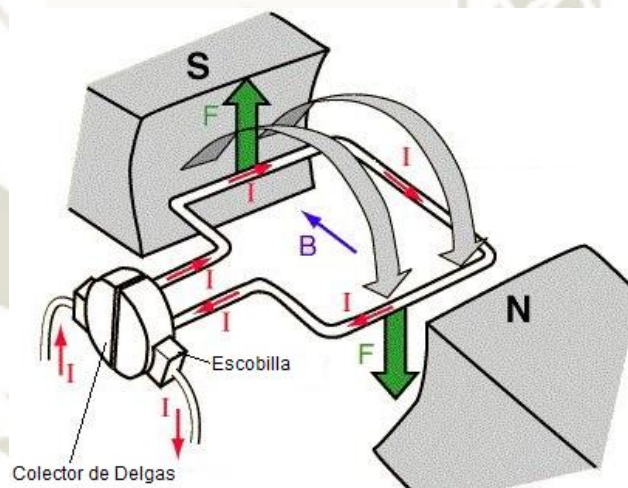


Figura 22. Funcionamiento del motor de corriente continua

Fuente: (Areatecnologia, 2018)

El motor visto en la Figura 23 consta de dos partes principales (Chapman, 2012):

- Posee un electroimán fijo llamado estator, el cual produce la fuerza necesaria sobre la espira.
- El conjunto de espiras se denomina rotor, representa la parte móvil.

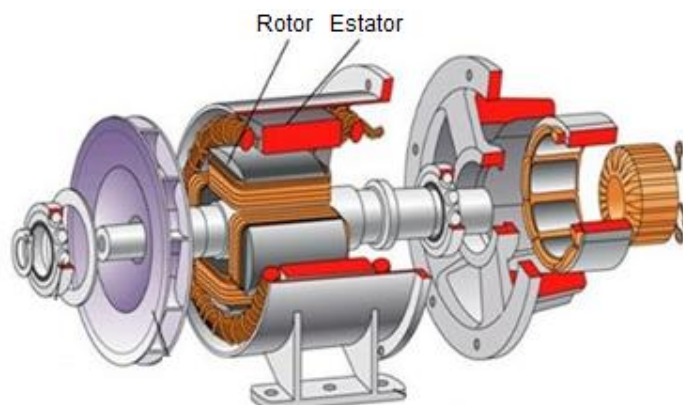


Figura 23. Partes del motor de corriente continua

Fuente: (Areatecnologia, 2018)

2.5.2 Motores de corriente alterna

Como su nombre lo indica son motores que se alimentan con corriente alterna. Existen varias clases de máquinas (Chapman, 2012):

- Monofásicos, vistos en la Figura 24.
 - Inducción.
 - Imán permanente.

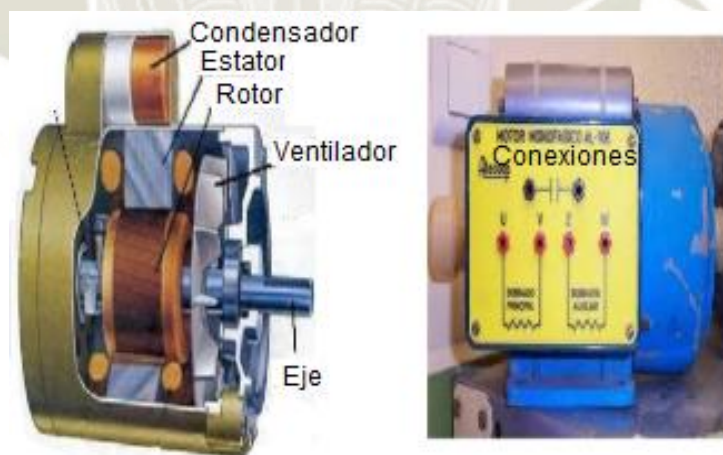


Figura 24. Motores de corriente alterna monofásico

Fuente: (Areatecnologia, 2018)

- Trifásicos, vistos en la Figura 25.
 - Asíncronos, rotores de jaula de ardilla y bobinado.
 - Síncrono: imán permanente, reluctancia, polos lisos y salientes.

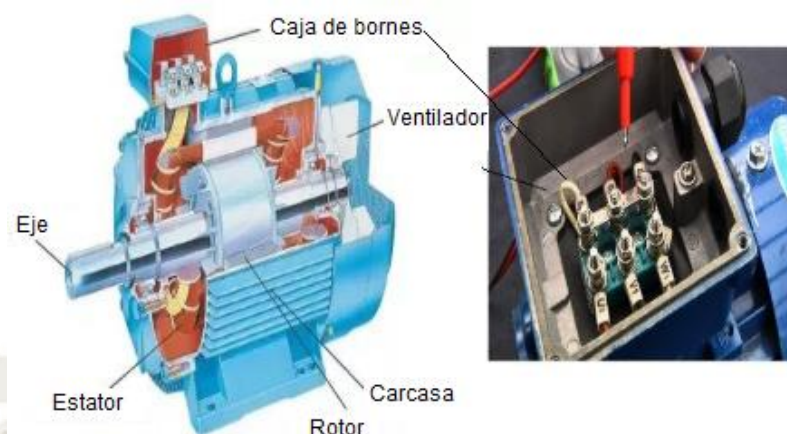


Figura 25. Motores de corriente alterna trifásico

Fuente: (Areatecnologia, 2018)

2.5.3 Servomotores

Existen varios tipos de servomotores, tales como: neumáticos, eléctricos, hidráulicos y digitales; su uso depende del tipo de aplicación debido a que cada uno representa un nivel de torque diferente o se utiliza cuando se requiere una alta precisión. A nivel industrial, los servomotores son motores eléctricos y pueden ser clasificados de la siguiente manera: motor AC (corriente alterna) y DC (corriente continua), vistos en la Figura 26 (Gandia, 2004).

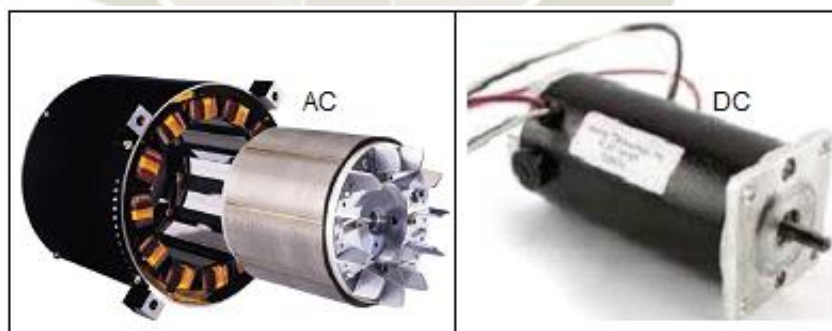


Figura 26. Servomotores AC y DC

Fuente: (Gandia, 2004)

Las partes internas del servomotor son (Areatecnologia, 2018):

- Motor eléctrico.
- Conjunto de engranes, a fin de regular la velocidad y torque del motor.

- Circuito electrónico que controla los movimientos del motor, estos son controlados mediante el uso de pulsos eléctricos (Figura 27). En la mayoría de los servomotores, no viene incluido este sistema de control, por lo que suele recurrirse al uso de un encoder.

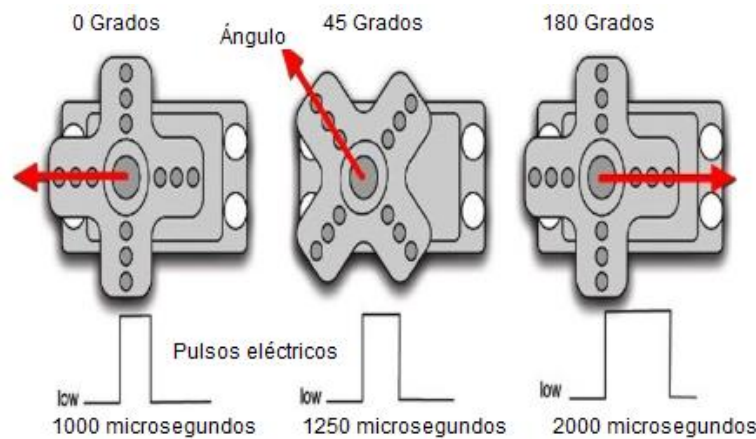


Figura 27. Control y giro de un servomotor

Fuente: (Areatecnologia, 2018)

- Y una resistencia variable (potenciómetro), la cual indica a que ángulo se encuentra el eje del motor.

2.5.4 Motor a pasos

Son motores compuestos por polos magnéticos (norte o sur) e imanes permanentes o bobinas de un devanado, visto en la Figura 28. Debido al gran número de polos, poseen una mayor precisión con respecto a los demás motores eléctricos, ya que como su nombre lo indica es un motor que gira mediante pasos (CLR, 2016).

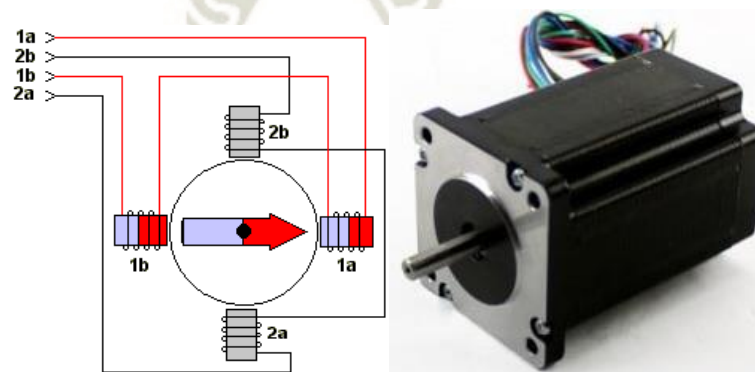


Figura 28. Motor a pasos

Fuente: (CLR, 2016), (330ohms, 2016)

Los motores a pasos pueden ser unipolares (5 o 6 cables) o bipolares (4 cables), cada uno posee diferentes características, vistos en la Tabla 7.

Tabla 7.

Características entre motores unipolares y bipolares

Descripción	M. Unipolar	M. Bipolar
Torque	Menor	Mayor
Anclaje	Menor	Mayor
Tamaño	Mayor	Menor
Precio	Alto	Bajo
Control	Sencillo (posee un circuito de alimentación)	Complejo (requiere de un control de giro y potencia)

Fuente: (330ohms, 2016)

2.6 Elementos de control

En todo proceso se sitúa un sistema de control o embebido que actúa como el cerebro del vehículo de superficie no tripulado y la inspección de calidad de agua. El cerebro del sistema de control puede ser operado por microcontroladores cuando se requiere utilizar en aplicaciones generales y/o microprocesadores para aplicaciones específicas. En la Figura 29, se indica una clasificación de estos elementos de control (Parra, 2012).

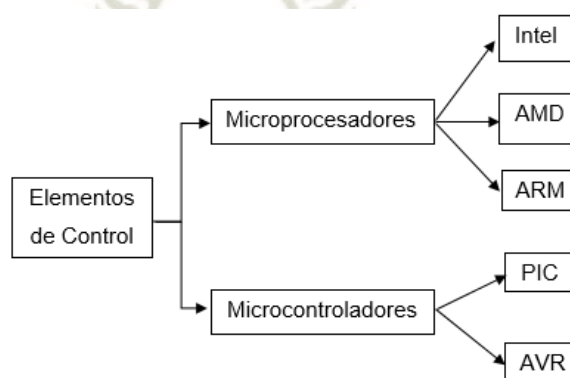


Figura 29. Elementos de control

Fuente: Elaboración Propia

En la Tabla 8 se describe características entre los microprocesadores y microcontroladores.

Tabla 8.

Características generales de los elementos de control

Características	Microprocesador	Microcontrolador
Flexibilidad	Máxima	Menor
Costo	Relativamente alto	Bajo
Software	Complejo	Simple
Piezas externas	Si requiere	No requiere

Fuente: (Parra, 2012)

2.6.1 Microprocesadores

El microprocesador por lo general se encuentra dentro del computador y es capaz de realizar operaciones lógicas / aritméticas (ALU). El microprocesador puede ser de las marcas Intel y AMD. Actualmente, cuando se requiere hacer aplicaciones con mayor productividad y entretenimiento móvil se da a lugar al uso de microprocesadores de la empresa ARM, por ejemplo, la placa Raspberry (Figura 30) posee este tipo de procesador. Esta plataforma electrónica posee las siguientes características (Parra, 2012), (Raspberrypi, 2018):



Figura 30. Raspberry Pi

Fuente: (Raspberrypi, 2018)

- Esta plataforma es similar a una computadora, ya que puede conectarse elementos como: monitor, teclado y mouse.
- Es de código abierto.
- Posee un procesador broadcom, puertos USB, HDMI, acceso a tarjeta microSD, Ethernet, pines de entrada/salida, memoria RAM y acceso a la conexión de una cámara.
- Utiliza lenguajes de programación Python, Tini basic, C, entre otros.
- Posee varias plataformas, vistas en la Tabla 9.

Tabla 9.

Especificaciones de la Raspberry Pi

Características	Modelos					
	Pi 1, A	Pi 1, B	Pi 1, B+	Pi 2, B	Pi 3, B	Pi 3, B+
Fuente de alimentación	5 V	5 V	5 V	5 V	5 V	5 V
Broadcom	CPU + GPU + DSP + SDRAM + puerto USB					
CPU	700 MHz	700 MHz	700 MHz	900 MHz	1.2 GHz	1.4 GHz
Juego de instrucciones RISC	32 bits	32 bits	32 bits	32 bits	64 bits	64 bits
Memoria SDRAM	256 MiB	512 MiB	512 MiB	1024 MiB	1024 MiB	1024 MiB
Puertos USB 2.0	1	2	2	4	4	4
Entradas-video	Conector MIPI CSI					
Salidas - video	Conector RCA, HDMI, interfaz DSI					

Salidas - audio	Conector de 3.5 mm, HDMI					
Almacenamiento integrado	SD/MMC/S DIO	SD/MMC/SDI O	MicroS D	MicroS D	MicroS D	MicroSD
Ethernet (RJ-45)	-	10/100	10/100	10/100	10/100	10/100
vía hub USB						/1000
WiFi		-			802.11n	802.11n/ ac
Bluetooth		-			4.1	4.2 BLE
Comunicación	SPI, I2C, UART	SPI, I2C, UART	SPI, I2C, UART	Bus HAT ID	Bus HAT ID	Bus HAT ID
Periféricos GPIO	8	8	8	17	17	17
Consumo energético: A/W	0.5 / 2.5	0.7 / 3.5	0.6 / 3.0	0.8 / 4.0	0.8 / 4.0	0.8 / 4.0

Fuente: (Raspberrypi, 2018)

2.6.2 Microcontroladores

Los microcontroladores PIC y AVR son chips de 8 bits disponibles comercialmente y rentables, ya que disponen de periféricos independientes, posee un rendimiento de bajo consumo, los diseñadores pueden realizar aplicaciones sencillas o complejas mediante el uso de lenguajes de programación (C, C++, Bascom, Java, etc.). Un microcontrolador está compuesto de puertos entrada/salida, memoria de programa y RAM, temporizadores, señales de reloj y comunicaciones, visto en la Figura 31 (Balbuena, 2017), (Parra, 2012).

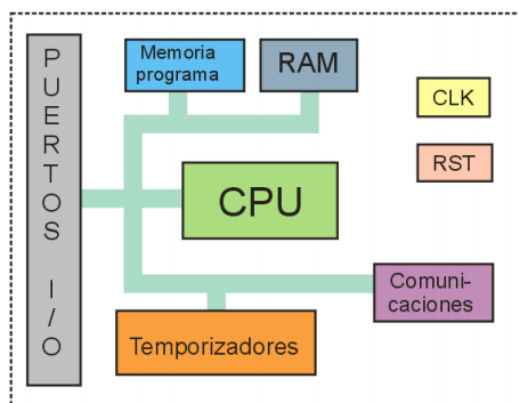


Figura 31. Bloques internos de un microcontrolador

Fuente: (Parra, 2012)

Actualmente se cuenta con la plataforma Arduino (Figura 32), que posee un microcontrolador de la familia AVR. Esta plataforma posee las siguientes características (Arduino, 2018):



Figura 32. Sistemas Embebidos

Fuente: (Arduino, 2018)

- Hardware y software de fácil acceso.
- Son de código abierto.
- Utiliza el lenguaje de programación y software basado en un entorno de desarrollo integrado IDE. Puede expandirse, añadiendo librerías C++ y/o usar código de ensamblador AVR-C.
- Son multiplataforma ya que puede ejecutarse en Mac, Windows y Linux.

- Las placas electrónicas de Arduino son económicas con respecto a otros sistemas embebidos.
- En la Tabla 10 se indican algunas plataformas de Arduino.

Tabla 10.

Especificaciones de los Arduinos

Características		Arduino						
		Nano	Uno	Leonardo	Due	Yún	Mega	Ethernet
Voltaje de funcionamiento [V]		5	5	5	3.3	5	5	5
Corriente DC [mA] en cada pin I/O		40	40	40	130	40	40	40
Pines I/O digitales		14	14	20	54	20	54	14
Pines entradas análogas		6	6	12	12	12	16	6
Canales PWM		6	6	12	12	7	15	4
Memoria flash [KB]		32	32	32	512	32	256	32
Sram [KB]		2	2	2	96	2.5	8	2
Eeprom [KB]		1	1	1	-	1	4	1
Velocidad de reloj [MHz]		16	16	16	84	16	16	16

Fuente: (Arduino, 2018)

Las fuentes de alimentación se encargan de suministrar un voltaje para el funcionamiento de los sensores, motores, circuitos de control y módulos electrónicos. Existen fuentes que parten de la red eléctrica AC y otras donde ya poseen energía almacenada llamadas baterías, visto en la Figura 33. Estas poseen tres características que las definen (UTFSM, 2010):

-
- This image displays a variety of industrial batteries. In the top row, from left to right, there is a white rectangular AGM battery with a blue top, a tall grey VRLA battery, a small brown flooded lead-acid battery, a tall red VRLA battery, and a tall light blue VRLA battery. In the bottom row, there is a large black flooded lead-acid battery on the left and a large black VRLA battery on the right, which is composed of four individual cells.

Fuente: (UTFSM, 2010)

Existen varios tipos de baterías, vistos en la Tabla 11.

Tabla 11.

Especificaciones de las baterías

Descripción – Tipos	Níquel		Litio		
	Plomo ácido	Ni-Cd	Ni-Mh	Li-ion	Li-Po
Energía/peso (Wh/kg)	30-50	48-80	60-120	110-160	100-130
Voltaje V	2	1.25		3.16	3.7
Número de recargas	1000	500	1000	4000	5000
Tiempo de carga en horas	8-16	10-14	2-4	2-4	1-1.5
Auto descarga por mes (%)	5	30	20	25	10

Fuente: (UTFSM, 2010)

2.7.1 Batería de plomo ácido

Estas baterías son utilizadas en los automóviles, carretillas, locomotoras, embarcaciones, en el área industrial como servicios continuos, regulación de cargas, energías renovables; es decir, se aplican en la automoción y en sistemas donde exigen continuidad de energía eléctrica. Además, cuenta con varias versiones (UTFSM, 2010):

- **Shallow-cycle (ciclo corto).** Normalmente usado en automóviles donde se requiere una mínima explosión de energía para encender el motor.
- **Deep-cycle (ciclo profundo).** Usado cuando se requiere ciclos repetitivos de carga y descarga. La mayor parte de aplicaciones utilizan este tipo de ciclo.
- **Gel-cell (sellada con aditivos).** Consiste en introducir el electrolito en un gel antiderrames, aunque este tipo de baterías posee un elevado costo, por lo que es utilizado en aviones militares.

Ventajas: son de bajo costo y tolerantes a un uso continuo (González, 2015).

Desventajas (González, 2015):

- La operación de la batería afecta la vida útil debido a la oxidación – reducción que se lleva a cabo, esto causa un deterioro de los electrodos.
- La temperatura de operación afecta la capacidad de almacenamiento y la vida útil.
- Cuando se tiene ciclos constantes, eso disminuye la capacidad de carga y puede ocasionar fallas en el suministro, afectando el rendimiento de la batería.

Debido a estos inconvenientes que presenta esta batería, se debe realizar un mantenimiento constante, revisión y reposición de los niveles de agua en el electrolito (González, 2015).

2.7.2 Batería de Níquel

Existen baterías de níquel – cadmio que poseen las siguientes características (UTFSM, 2010):

- Posee una baja tasa de auto descarga, visto en la Figura 34.

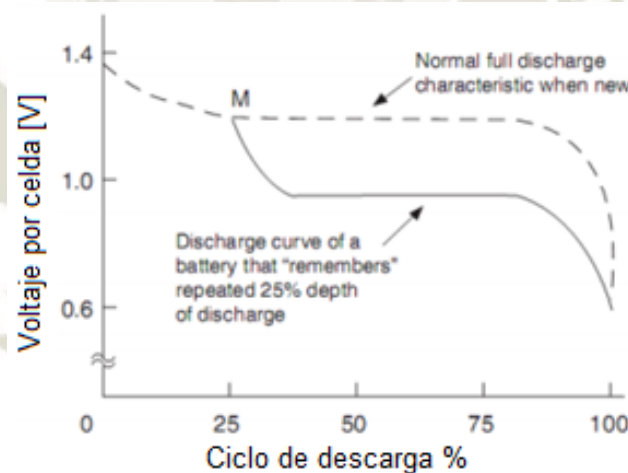


Figura 34. Voltaje vs ciclo de descarga

Fuente: (UTFSM, 2010)

- Contiene celdas selladas debido a la tolerancia en altas temperaturas.
- Puede acelerar su proceso de descarga debido al efecto de memoria que posee.

Ventajas (González, 2015):

- Requieren mantenimientos con menor frecuencia.

- Tienen la capacidad de entregar corriente en el menor tiempo posible.
- Posee una vida útil superior a las baterías de plomo ácido.
- Puede mantener un ciclo de carga/descarga profunda.
- Tiene una resistencia interna baja y gracias a ello se puede tener conexión de baterías en serie.

Desventajas (González, 2015):

- Alto costo en comparación a las baterías de plomo ácido.
- Alta toxicidad del cadmio.
- Pierden su capacidad cuando no se descargan completamente la batería.

También existen baterías de níquel – hidruro metálico (Ni-Mh) el cual es una tecnología implementada luego de la batería Ni-Cd. Posee las siguientes ventajas y desventajas (UTFSM, 2010):

- Tiene mayor densidad de energía.
- Evita problemas ambientales debido a que el ánodo está formado por el metal hidruro.
- No puede entregar altos picos de potencia.
- Es peligrosa en el caso de que existe sobrecarga de la batería.
- Costo elevado.
- Tiene una tasa alta de auto descarga.

Usualmente este tipo de baterías es utilizado en (UTFSM, 2010):

- Los vehículos de propulsión eléctrica o híbridos.
- Robots de tipo humanoides diseñado por Honda.

2.7.3 Batería de litio

Existen baterías de ion-litio que poseen las siguientes aplicaciones, ventajas y desventajas, vistas en la Tabla (UTFSM, 2010):

Aplicaciones:

- Industria automotriz, para automóviles eléctricos.
- Gestión de energía.
- Dispositivos portátiles como teléfonos móviles, laptops, lectores de música, etc.

Ventajas (González, 2015):

- Tamaño menor en comparación a las demás baterías.
- Alta densidad de energía.
- Alta capacidad de carga/descarga.
- Resistencia interna baja.

Desventajas (González, 2015):

- Se reduce la vida útil cuando no hay carga en la batería.
- El costo depende de la aplicación a utilizarse, ya que cada celda posee 3.5 Voltios.
- Sensible a las temperaturas altas.
- Es peligroso ya que puede producirse una explosión o incendio de la batería.

También existen baterías de polímero – litio (Li-Po), cuyas ventajas y desventajas son las siguientes (UTFSM, 2010):

- Debido a su constitución física, el polímero sólido actúa como electrolito.
- Poseen una densidad de energía entre 5 a 12 veces que las baterías de Níquel a un mismo peso.
- Tienen la misma capacidad, pero 4 veces más ligera que las baterías de Ni-Cd.
- Cada batería cuenta con un voltaje nominal de 3.7 V. Debido a ello, no se debe descargar por debajo de los 3 V y cargar por encima de los 4.3 V por celda.
- Requiere de un mantenimiento especial para evitar deterioro de la batería, a fin de evitar que se produzca una ignición o explosión de esta.

2.8 Medios de comunicación

Los medios de comunicación sirven de ayuda para la transmisión y recepción de datos desde un punto a otro, visto en la Figura 35. Las redes de comunicación pueden ser de forma alámbrica e inalámbrica.

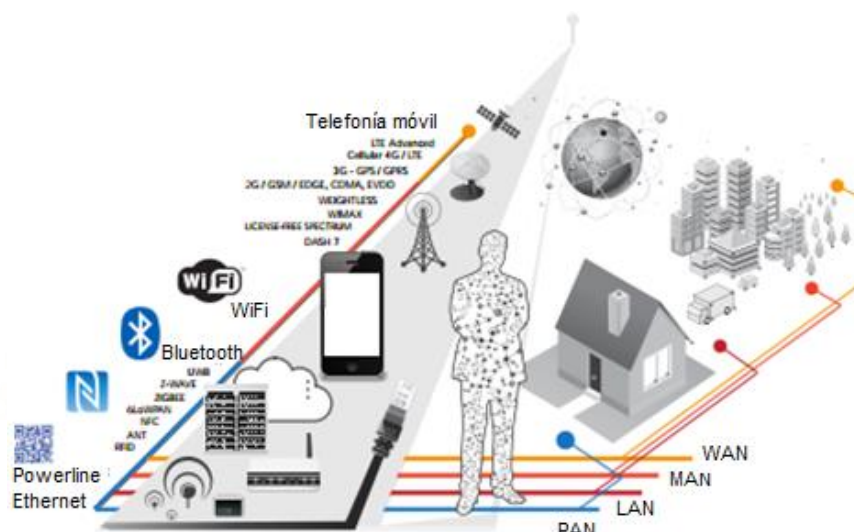


Figura 35. Medios de comunicación

Fuente: (Sánchez C. , 2016)

En la Tabla 12, se describe de manera general las características de las redes alámbricas e inalámbricas.

Tabla 12.

Cuadro comparativo de redes de comunicación

Comunicación	Características
Alámbrica	<ul style="list-style-type: none"> Se conecta mediante cables UTP. Utiliza conectores RJ-45. Se debe conocer el tipo de infraestructura y diseñar la ruta de cableado.
Inalámbrica	<ul style="list-style-type: none"> No necesita cables para la conexión de la red. Dependiendo del tipo de red, se evalúa la distancia máxima de la transmisión/recepción de datos. Se debe tener un emisor y receptor.

Fuente: (Proaño, 2009)

2.8.1 Comunicación alámbrica o cableada

Este tipo de redes utiliza cable de datos para conectar computadores y otros dispositivos entre sí. La velocidad de transmisión depende del tipo de cable (par trenzado, coaxial y fibra óptica). A continuación, en la Tabla 13 se describe las características de los tipos de cable (Ies, 2018).

Tabla 13.

Tipos de cable

	Par trenzado	Coaxial	Fibra óptica
Tipo de señal	Corriente eléctrica		Luz (onda electromagnética)
Atenuación	Considerable	Intermedia	Baja
Interferencias	Considerable	Baja	Baja
Ancho de banda	Bajo / Intermedio	Alto	Muy alto
Costo	Bajo	Intermedio	Alto
Distancia	Corta	Media / Larga	Larga / Muy larga

Fuente: (Ies, 2018)

Los microcontroladores y microprocesadores utilizan los siguientes estándares de comunicación alámbrica: RS-232, RS-485, RS-422, usb y ethernet.

Comunicación RS-232 (Forero, 2012)

- La comunicación es de tipo full dúplex.
- Cuando utiliza un módem, se conecta mediante un conjunto de cables, visto en la Figura 36.

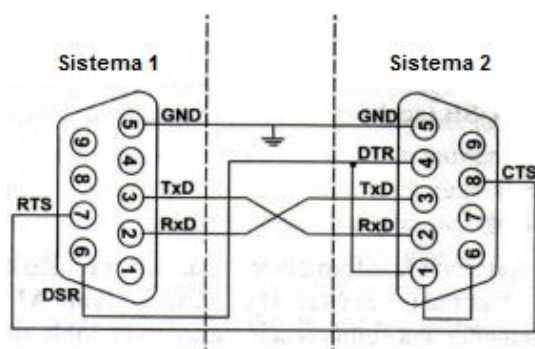


Figura 36. Conector serial D9

Fuente: (Forero, 2012)

- Intercambia información entre dos puntos de un sistema a nivel local.
- En la actualidad, es superado por el estándar USB.

Comunicación RS-422 (Forero, 2012)

- La transmisión se realiza mediante cables de par trenzado de cobre y terminales RJ.
- Son resistencias al ruido.
- La velocidad de transmisión es superior al estándar RS-232.
- La comunicación es de tipo full y half dúplex.
- Es compatible con el estándar RS-232 ya que dispone de un adaptador de 8 puertos y terminales 232, visto en la Figura 37.



Figura 37. Estándar RS-422

Fuente: (Forero, 2012)

Comunicación RS-485 (Forero, 2012)

- Sus características son similares al estándar RS-422 referentes al conexionado.
- Permite la conexión hasta 32 emisores y receptores.

- La comunicación es de tipo half dúplex.
- En la actualidad, es superado por el estándar Ethernet.

Como puede visualizarse cada comunicación posee características similares o de mayor capacidad, tal como se indica en la Tabla 14.

Tabla 14.

Estándares RS-232, RS-422 y RS-485

	RS-232	RS-422	RS-485
Emisores / receptores	1 emisor 1 receptor	1 emisor 10 receptor	32 emisor 32 receptor
Modo de comunicación	Full dúplex	Full dúplex Half dúplex	Half dúplex
Distancia máxima	15 mts a 19,2 Kbps	1,2 km a 100 Kbps	1,2 km a 100 Kbps
Corriente de salida	500 mA	150 mA	250 mA
Velocidad / Distancia	Mínimo	-	100 kbps / 1,2 km
	Medio	-	1 Mbps / 120 mts
	Máximo	20 kbps / 15 mts	10 Mbps / 12 mts
Máximo voltaje a la línea de salida	± 25 V	-0.25 a +6 V	-7 a 12 V

Fuente: (Forero, 2012)

Bus serial universal USB (Hurtado, 2012)

- Posee diferentes tipos de conexión entre el computador y periféricos con el propósito de eliminar las conexiones serie y paralelo.
- La conectividad es limitada, no supera los 5 metros.

Ethernet (Hurtado, 2012)

- Es un potente estándar para la transmisión y recepción de datos.
- Utiliza el protocolo de comunicación TCP/IP.
- Tiene una alta confiabilidad e integridad de la red.

2.8.2 Comunicación inalámbrica

Los medios de transmisión/recepción de datos se lleva a cabo mediante antenas en la comunicación inalámbrica. Existen varios tipos de comunicación, pero se va a tratar los que normalmente se utilizan en las plataformas Arduino y Raspberry Pi, tales como: wifi, zigbee/xbee, bluetooth, gsm, umts, etc., vistos en las Tablas 15 y 16 (Aprendiendoarduino, 2018).

Tabla 15.
Medios de comunicación

Características	Zigbee	WiFi	Bluetooth
Estándar IEEE	802.15.4	802.11n	802.15.1
Frecuencia GHz	2.4	2.4 y 5.0	2.4
Alcance (metros)	10 – 100	50	3 – 150
Velocidad de transferencia (Mbps)	0.25	0.15 – 1.0	1.0

Fuente: (Aprendiendoarduino, 2018)

Existen tecnologías de segunda (chips GSM), tercera generación (UMTS) y GPRS (entre segunda y tercera generación) e híbridas GSM/GPRS que poseen las siguientes características (Camargo, 2009):

Tecnología GSM (sistema global para las comunicaciones móviles)

- Posee una transferencia de datos igual a 9.6 kb/s.
- Utiliza bandas con licencias.
- El área de cobertura es limitada, debido a la reutilización de frecuencias.

- Posee una velocidad aproximada a los 200 km/h.
- Las zonas de espectro son 900/1800 MHz.

Tecnología GPRS (servicio general de paquetes vía radio)

- Es compatible con la tecnología GSM.
- La velocidad de transmisión oscila entre los 54 a 172 kb/s.

Tecnología UMTS (sistema universal de telecomunicaciones móviles)

- Utiliza CDMA en vez de TDMA. Gracias al cdma, utiliza un tipo de código.
- Proporciona una reutilización de frecuencias totales.
- La velocidad de transmisión varía de 144 kb/s a 2 Mb/s.
- La banda de frecuencias es cercana a 2 GHz.

Tecnología GSM/GPRS

- Sirve para protección de circuitos contra sobrecargas, cortocircuitos, etc.
- Posee varios puertos de comunicación RS-232.

Capítulo III

3. DISEÑO DEL SISTEMA

El vehículo de superficie no tripulado consta de mecanismos y dispositivos electrónicos para la evaluación de los parámetros de calidad del agua. A continuación, se observa los diferentes diseños mecánicos y electrónicos.

3.1 Diseño del vehículo de superficie no tripulado

3.1.1 Selección de la forma del vehículo

Para el inicio del diseño mecánico se busca un modelo de barco a escala que cumpla con las condiciones de movimiento y estabilidad necesaria para el ensayo requerido. Este modelo debe proporcionar un eficaz rendimiento de los espacios, aumentando el área útil de la embarcación. Inicialmente se toma en cuenta la base del vehículo de superficie no tripulado, que está constituida de las siguientes partes: codaste, baos, trancanil, quilla y roda, visto en la Figura 38 (Rebassa & Castells, 2016)

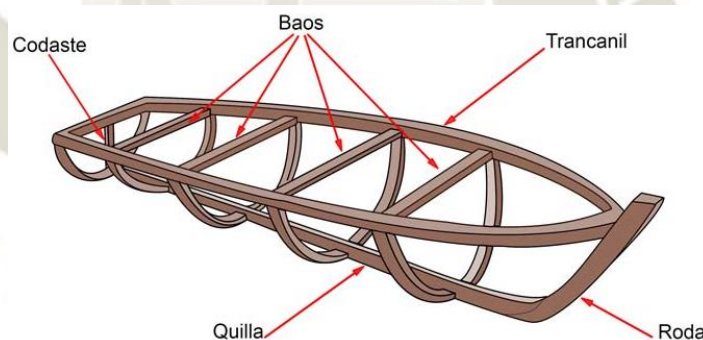


Figura 38. Base del vehículo de superficie no tripulado

Fuente: Elaboración Propia

Se tiene especial cuidado en la quilla, la cual otorga buena estabilidad a cualquier velocidad, pero su principal ventaja es que cuando se aplica suficiente potencia, puede llegar a planear. Por otra parte, para la investigación de este trabajo se necesita estabilidad estructural, ya que se cuenta con componentes electrónicos que añaden un peso extra y deben poseer una protección para su funcionamiento. En la Figura 39 se muestra el vehículo de superficie no tripulado a utilizar, el mismo que es un Aerodeslizador Anfibio tipo tanque.



Figura 39. Vehículo de superficie no tripulado

Fuente: (AliExpress, 2018)

3.1.2 Velocidad del bote

El servomotor posee las siguientes características técnicas:

- Voltaje: 7.2 V.
- Corriente: 350 mAh.
- Rapidez: 574.2 rpm, visto en la Figura 94 del Capítulo V.
- Diámetro del eje: 40 mm, visto en la Figura 93 del Capítulo V.

En la Tabla 16, se observa el cálculo de la velocidad tomando como base la rapidez de 576.6 revoluciones por minuto y un diámetro de eje de 40 milímetros.

Tabla 16.

Velocidad angular y lineal

Magnitud	Valor	Unidad de medida
Diámetro del eje	0.04	m
Velocidad angular	574.2	rpm
	60.13	rad/s
Velocidad lineal	2.41	m/s

8.66

km/h

Fuente: Elaboración Propia

3.1.3 Estudios de formas para una embarcación

El principio que se emplea en el diseño de barcos o modelos a escala de embarcaciones es el principio de Arquímedes, el cual dice que “todo cuerpo total o parcialmente sumergido en un fluido en reposo experimenta un empuje vertical hacia arriba igual al peso del volumen del fluido desalojado” (Rebassa & Castells, 2016)

La capacidad de flotación que posee un barco se presenta gracias a la presión hidrostática que sufre el casco, ya que, con una velocidad igual a cero, el balance de fuerzas hidrostáticas es igual al peso del volumen del agua arrojada. Dicho en otra forma, una vez inicia el movimiento, existe una fuerza hidrodinámica que inicia la aceleración de las partículas de agua cercanas, las cuales ejercen una fuerza de igual dirección y sentido que la producida por la misma placa. En la Figura 40 se observa el balance de fuerzas ejercidas sobre una placa plana. La fuerza resultante es la responsable del levantamiento de la placa y del ángulo de trimado de la embarcación (Rebassa & Castells, 2016).

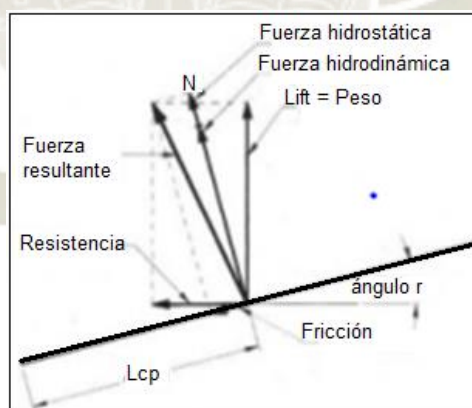


Figura 40. Balance de fuerzas sobre una placa plana

Fuente: (Larsson & Eliasson, 2000)

Se presenta una presión hidrodinámica que se incrementa a medida que se eleve la velocidad de la placa, levantándola hasta alcanzar la condición de planeo; lo que se conoce como fenómeno de sustentación. Para definir las fronteras entre una embarcación de desplazamiento, semi desplazamiento o planeo, se recurre a la interpretación del

número de Froude de cada embarcación, visto en la Ecuación 6. Éste, es un número adimensional que relaciona las fuerzas de inercia y las fuerzas de gravedad que actúan sobre un fluido (Rebassa & Castells, 2016)

$$F_n = \frac{V}{\sqrt{g \cdot L}} \quad (6)$$

Donde:

- F_n es el número de Froude.
- V es la velocidad en metros sobre segundo (m/s).
- g es la gravedad igual a 9.81 (m/s²).
- L es la longitud del vehículo de superficie no tripulado (Figura 41) en metros.

En la Tabla 17 se describe la representación del valor obtenido de F_n .

Tabla 17.

Régimen de la embarcación

Régimen	Valor F_n	Descripción
Desplazamiento	$0 \leq F_n \leq 0.6$	Sustentación es despreciable.
Semi desplazamiento	$0.6 < F_n \leq 1.3$	El centro de gravedad recupera su posición inicial debido a la contribución de las fuerzas hidrodinámicas.
Planeo	$1.3 < F_n$	La mayor fuerza hidrodinámica provoca que el centro de gravedad se sitúe en valores positivos. El planeo total se da cuando la sustentación dinámica es igual al peso de la embarcación, percibiendo que la embarcación se desliza sobre el agua.

Fuente: (Rebassa & Castells, 2016)

A continuación, en base a la Ecuación 6 se calcula F_n :

$$F_n = \frac{2.41 \text{ m/s}}{\sqrt{(9.81 * 0.32) \text{ m/s}}} = 1.36$$

Eso quiere decir que F_n está en el régimen de planeo.

3.1.4 Resistencia al avance y velocidad del viento

Se trata de la fuerza que ejerce el agua sobre el casco en sentido contrario al avance del buque y que debe ser igualada para mantener una velocidad; se calcula mediante la Ecuación 7 (Rebassa & Castells, 2016).

$$F = L * B \quad (7)$$

Donde:

- L representa la eslora, es decir, el largo de la embarcación que es de 0.32 metros.
- B representa la manga máxima, es decir, el ancho máximo de la embarcación fija que es de 0.0675 metros (sin considerar el espesor de los lados).

$$F = 0.32 * 0.0675 = 0.0216$$

La fuerza del viento se encuentra relacionada mediante la escala de Beaufort, donde los vientos se ordenan según su velocidad, en la Tabla 18 se observa la velocidad y el nombre de viento según la escala Beaufort (SolerPalau, 2015).

Tabla 18.

Velocidad del viento en base a la escala Beaufort

Escala de Beaufort	Nombre de Viento	Velocidad	
		m/s	km/h
0	Calma	0.5	2
1	Aire ligero	1.5	5
2	Brisa ligera	3	11
3	Brisa suave	6	22
4	Brisa moderna	8	30
5	Brisa fresca	11	40

6	Brisa fuerte	14	50
7	Viento moderado	17	60
8	Viento fresco	21	75
9	Viento fuerte	24	87
10	Gran viento	28	100
11	Tempestad	32	115
12	Huracán	36 o más	130 o más

Fuente: (SolerPalau, 2015)

3.1.5 Centro de gravedad

Este parámetro sirve en una embarcación para que el arreglo sea considerado como perfecto, es decir que el agua se sitúe sobre la línea de flotación prevista, en donde el fenómeno se presenta cuando el centro de gravedad CG está vertical del centro longitudinal de empuje CE, siendo necesario el cálculo de este. En la Figura 41 se indican el centro de gravedad y longitudinal en la misma línea de acción vertical (Guerrero, 2011).

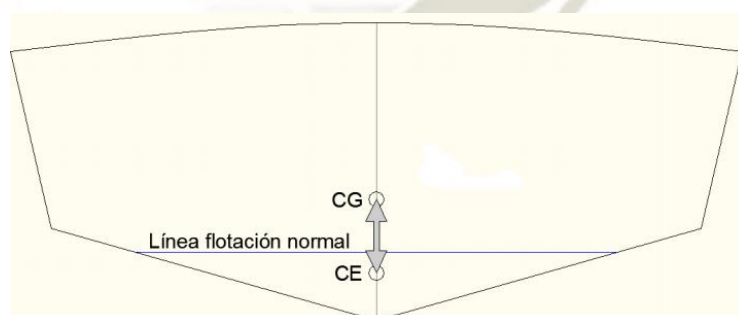


Figura 41. Centro de gravedad

Fuente: Elaboración Propia

El centro de gravedad se encontrará al dibujar el modelo en el programa SolidWorks, para realizar la simulación y el análisis de la incidencia del nuevo diseño en la hidrodinámica del vehículo de superficie no tripulado.

3.1.6 Diseño en el software CAD

Como se explicó en el apartado 3.1.1., el modelo seleccionado para la realización del prototipo es un barco tipo anfibio (Figura 39). Por otra parte, para la implementación de los componentes electrónicos explicados en otras secciones, se extrae la parte superior, teniendo así alojamientos y un punto de inicio de diseño mecánico, tal como se indica en la Figura 42. Este es un requerimiento esencia para iniciar con el proceso de medición para su dibujo en el software CAD denominado SolidWorks 2017.

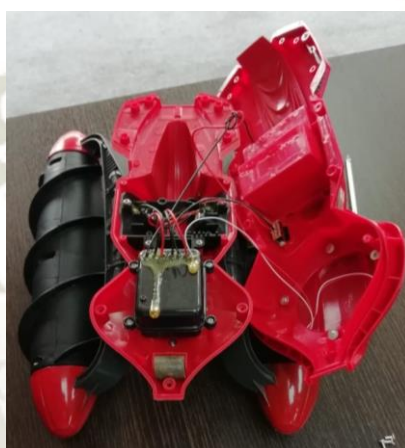


Figura 42. Extracción de la carcasa superior

Fuente: Elaboración Propia

3.1.7 Dibujo en el programa SolidWorks 2017

Se procede a dividir el vehículo de superficie no tripulado (Figura 43) en tres partes:

- (1) es la base.
- (2) donde se ubican los motores.
- (3) representa la tapa superior del vehículo.

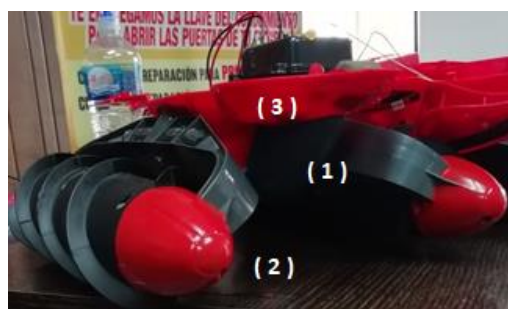


Figura 43. Distribución de partes del vehículo

Fuente: Elaboración Propia

Tapa superior y base del vehículo

El diseño se inicia con la medición de la tapa superior, en donde se observan las cavidades que se pueden usar como alojamientos de los sistemas electrónicos. En la Figura 44 se observa la vista superior de la tapa con sus respectivas medidas.

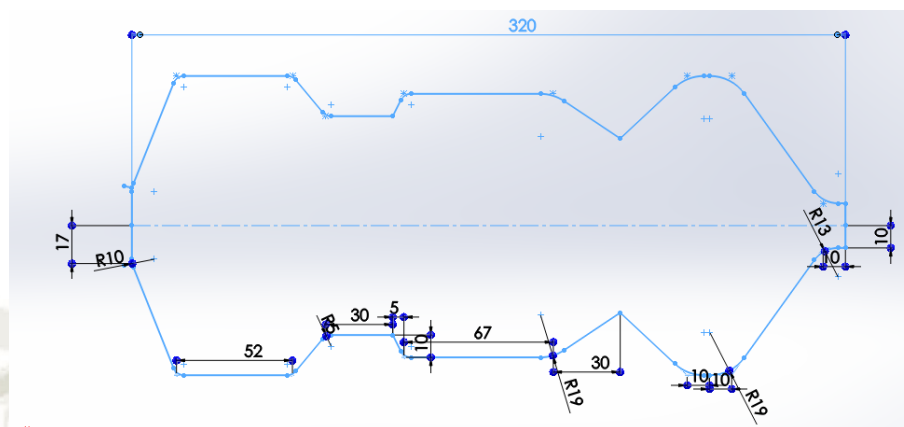


Figura 44. Vista superior de la tapa

Fuente: Elaboración Propia

Posteriormente, se dibuja la base del barco y se une al croquis de la tapa superior. Este paso es necesario, ya que con el mismo se efectúa un solo vaciado de 3 milímetros (mm). Realizando un sólido con la tapa y base, se garantiza que no habrá errores cuando se realice el ensamble con los motores, visto en la Figura 45.

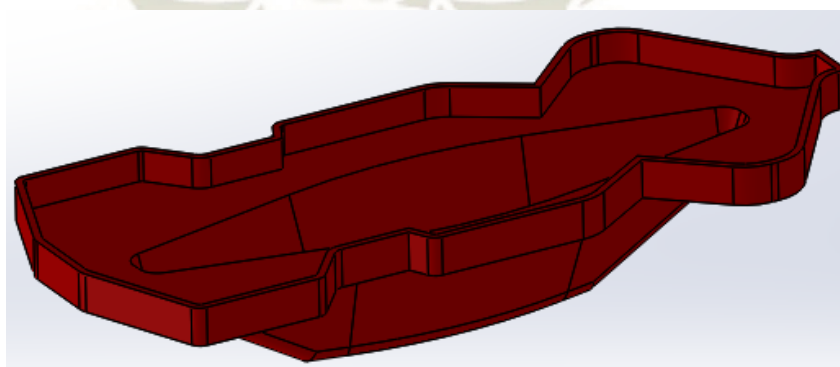


Figura 45. Sólido de la base y tapa del vehículo

Fuente: Elaboración Propia

Motores y tornillo sin fin del barco

Para que exista movimiento se dibuja en el software los tornillos sin fin, en donde se encuentran integrados los servomotores de corriente continua, visto en la Figura 46.

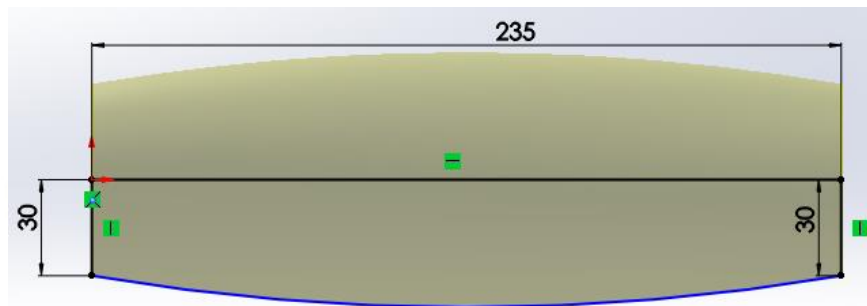


Figura 46. Medidas iniciales para el cuerpo del tornillo

Fuente: Elaboración Propia

Finalmente, se realiza una hélice en el cuerpo, tal como se indica en la Figura 47.



Figura 47. Hélice para el servomotor

Fuente: Elaboración Propia

Ensamble inferior

Antes de integrar los componentes electrónicos, se realiza el ensamble inicial inferior del prototipo, visto en la Figura 48.

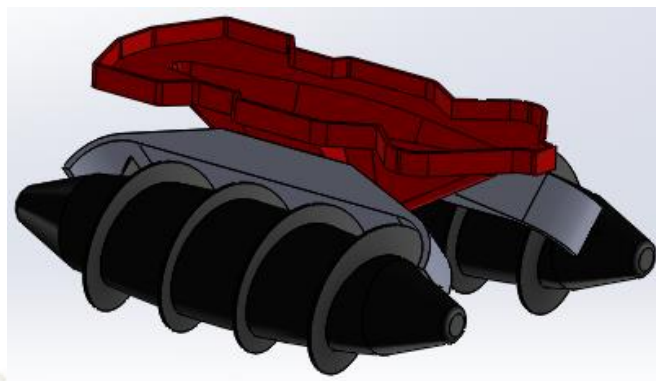


Figura 48. Ensamble inferior

Fuente: Elaboración Propia

Ensamblaje superior

- **Tapa superior delantera**

La parte superior del bote (Figura 50) fue rediseñada para que se pueda instalar el sistema electrónico. Por lo tanto, se dividirá en dos tapas para la protección del sistema cuando se encuentre en operación. En la Figura 49 se muestra la tapa delantera dibujada en el software con sus respectivas medidas.

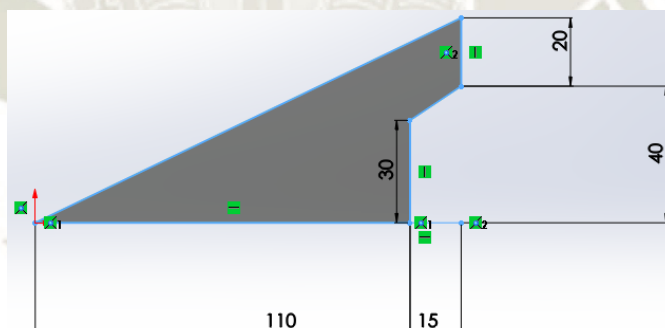


Figura 49. Medidas de la tapa superior delantera

Fuente: Elaboración Propia

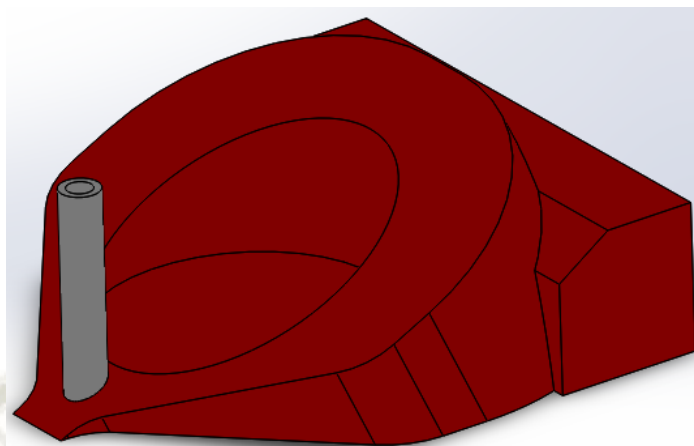


Figura 50. Tapa superior delantera

Fuente: Elaboración Propia

- **Tapa superior trasera**

Es la parte superior del barco (Figura 52) que cubre los componentes electrónicos necesarios para el funcionamiento de la investigación, se encuentra 2 centímetros sobre la base inferior, la cual sirve como soporte de las partes y que no haya entrada de agua cuando se encuentre en funcionamiento. En la Figura 51 se observa las medidas de la tapa posterior.

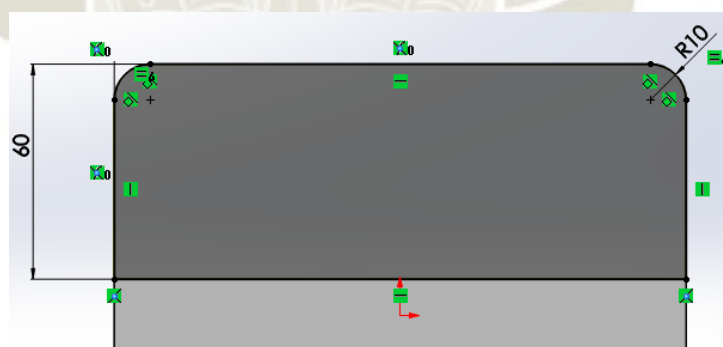


Figura 51. Medidas de la tapa superior trasera

Fuente: Elaboración Propia

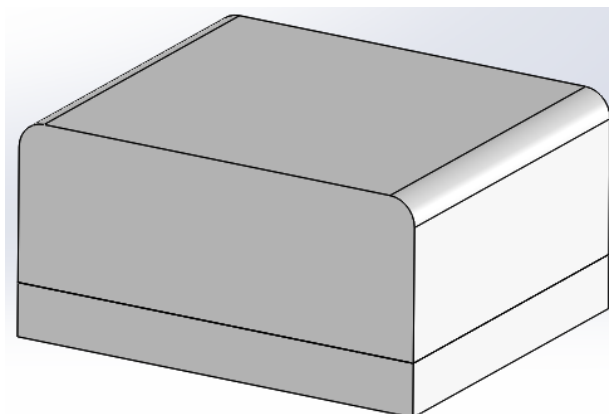


Figura 52. Tapa superior trasera

Fuente: Elaboración Propia

Acoples para la extracción de agua

Para la extracción del agua, se implementa un sistema de bombeo que extrae el agua del sitio donde esté ubicado el vehículo de superficie no tripulado a un vaso que es colocado en la cavidad superior de la tapa frontal. Se han colocado dos codos de $\frac{1}{2}$ pulgada y dos acoples que dirigen el fluido desde la manguera al vaso. El ensamble final se muestra en la Figura 53.

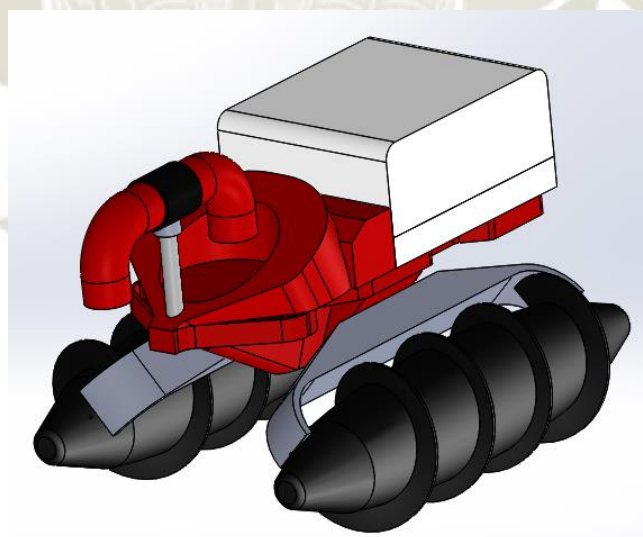


Figura 53. Vehículo de superficie no tripulado final

Fuente: Elaboración Propia

3.2 Esquemas eléctricos del conexionado de sensores y actuadores con los elementos de control

En las Figuras 54 y 55 se observan los esquemas eléctricos del sistema formado por sensores, controladores y actuadores. El primero representa la conexión hacia el Arduino Nano y el segundo indica la conexión hacia la Raspberry Pi3 B+.

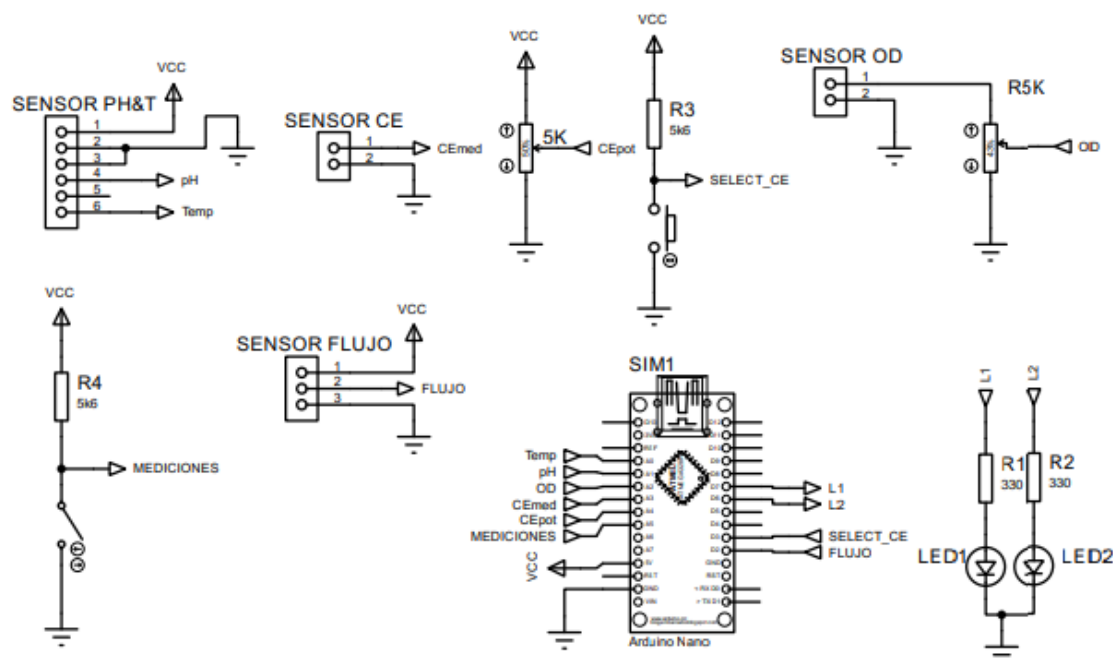


Figura 54. Circuito electrónico con el Arduino Nano

Fuente: Elaboración Propia

Además, se tiene conexión mediante cables y buses de datos de los siguientes dispositivos:

- Fuente de alimentación de 5V a la tarjeta Raspberry.
- Fuente de alimentación de 7.2V a los motores.
- Tarjeta Raspberry Pi3 B+ a la plataforma Arduino Nano.
- Sensores, actuadores y controladores.
- Cámara a la Raspberry Pi3 B+.

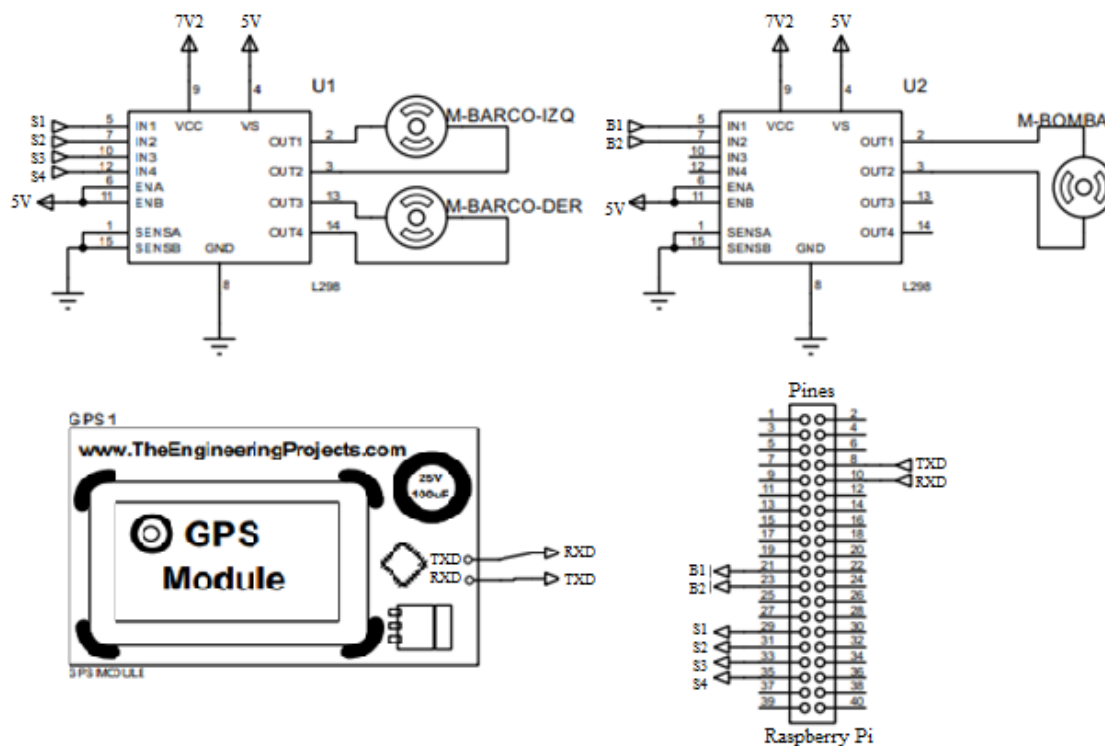


Figura 55. Circuito electrónico con la Raspberry Pi3 B+

Fuente: Elaboración Propia

3.3 Selección de sensores

3.3.1 Sensor para la medición de pH y temperatura

Para las mediciones de pH y temperatura se opta por escoger un sensor comercial de la marca Gaohou ph0-14, el cual posee las siguientes características (Mokey, 2018):

- Tiene una sonda de electrodo de pH con conector BNC, visto en la Figura 56.



Figura 56. Sonda de electrodo de pH

Fuente: (Mokey, 2018)

- Posee un circuito de acondicionamiento que mide los valores de detección pH y temperatura por medio de señales análogas, visto en la Figura 57 (Mokey, 2018).

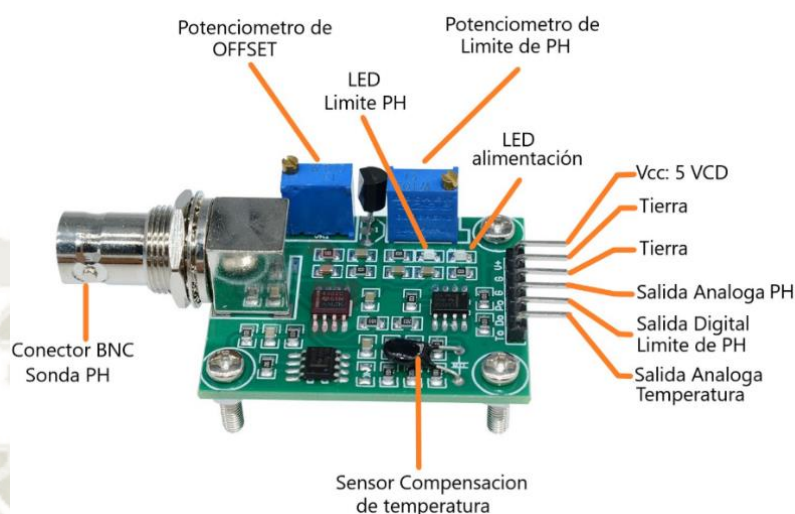


Figura 57. Circuito de acondicionamiento para el sensor pH y temperatura

Fuente: (Diymore, 2018)

- En la Tabla 19, se destacan las siguientes características técnicas de operación.

Tabla 19.

Características técnicas de la sonda de pH

Características	Valor	Unidad de Medida
Fuente de alimentación	5 ± 0.2	Vac o Vdc
Corriente de trabajo	5 – 10	mA
Detección de pH	0 – 14	pH
Detección de temperatura	0 – 80	°C
Tiempo de respuesta	≤ 5	Segundos
Tiempo - establecimiento	≤ 60	Segundos
Potencia	≤ 0.5	Vatios W
Humedad nominal	65	% RH

Fuente: (Diymore, 2018)

3.3.2 Sensor para la medición de conductividad eléctrica

Para la medición de conductividad eléctrica (Figura 58) se opta por escoger un medidor inductivo Va 1022, el cual posee las siguientes características (Vimael, 2018):

- Permite trabajar en lugares oscuros.
- Posee un circuito para compensación de temperatura.
- El medidor posee protección del equipo con la norma IP65, es decir protección contra el polvo y chorros de agua en todas las direcciones.
- Para limpieza de los electrodos se debe utilizar agua destilada.
- En la Tabla 20, se describen las características técnicas del medidor.

Tabla 20.
Conductímetro Marca VA 1022

Características	Valor	Unidad de Medida
Fuente de alimentación	3	Vdc
Rango	0 – 1999	us/cm
Resolución	1	us/cm
Precisión	± 2	%
Compensación de temperatura	0 – 70	°C

Fuente: (Vimael, 2018)



Figura 58. Medidor de conductividad eléctrica VA 1022

Fuente: (Vimael, 2018)

El medidor dispone de varios circuitos de tipo SMD (dispositivos de montaje superficial) y una placa de doble lado, visto en la Figura 59 lo cual dificulta verificar su modo de funcionamiento.

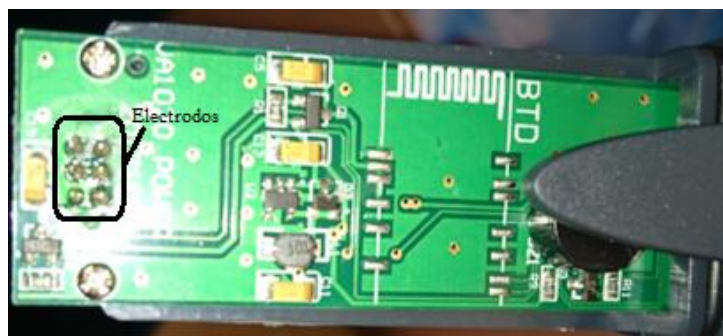


Figura 59. Circuito interno del medidor de conductividad eléctrica

Fuente: Elaboración Propia

Por motivos de costos del medidor, se ha empleado varios modos de operación para la respectiva medición, vistos en la Tabla 21 y Figura 60.

Tabla 21.

Modos de operación

Operación	Descripción
0 0	No se realiza medición
0 1	Medición con potenciómetro
1 0	Medición de agua de la llave
1 1	Medición de agua con azúcar u otra mezcla (Se agregaría la ecuación en la plataforma Arduino Nano)

Fuente: Elaboración Propia

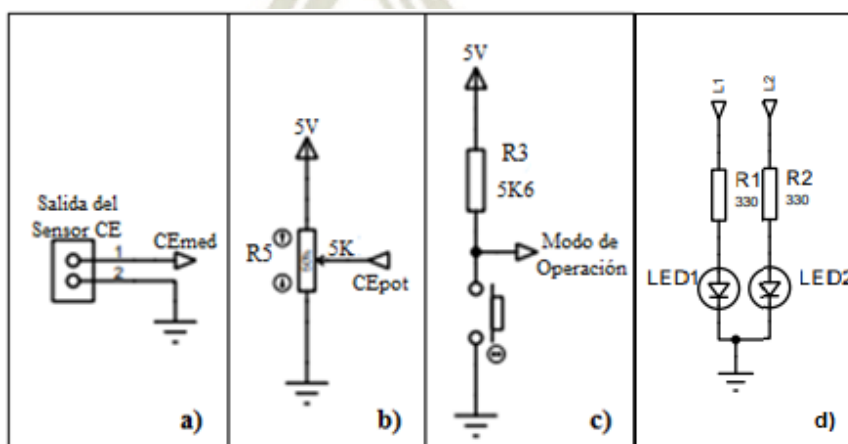


Figura 60. Diagrama circuital para la medición de conductividad eléctrica

Fuente: Elaboración Propia

Donde:

- La Figura 60.a) representa los dos puntos de salida del sensor de conductividad eléctrica de la Figura 58 (medidor real).
- La Figura 60.b) representa el potenciómetro con el cual se indica la medición del medidor. La resistencia se establece con la Ley de Ohm, visto en la Ecuación 8 y el valor de corriente (40 mA) se escoge de la Tabla 9.

$$R_{\min} = \frac{V}{I_{\text{Arduino}}} \quad (8)$$

$$R_{\min} = \frac{5V}{40 \text{ mA}} = 125\Omega$$

Se asume un potenciómetro logarítmico de 5 k Ω para la obtención de una baja corriente y no sobrecargar el pin de entrada al Arduino Nano.

$$I_{\text{CEpot}} = \frac{5V}{5\text{k}\Omega} = 1\text{mA}$$

- La Figura 60.c) indica el modo de operación para la medición de conductividad eléctrica. Se asume una resistencia de 5.6 k Ω y la corriente que entra al Arduino es igual a:

$$I_{\text{Modo-Operación}} = \frac{5V}{5.6\text{k}\Omega} = 0.89\text{mA}$$

- La Figura 60.d) posee indicadores del modo de operación para la medición de conductividad eléctrica. Se asume una resistencia de 330 Ω y la corriente que entra al Arduino es igual a:

$$I_{\text{Led}} = \frac{5V - 1.7 \text{ (voltaje del led)}}{330 \Omega} = 10 \text{ mA}$$

3.3.3 Sensor para la medición de oxígeno disuelto

Para la medición del oxígeno disuelto (Figura 61) se opta por escoger un medidor amperométrico de la marca Pinpoint II, el cual posee las siguientes características (Marine, 2018):

- Fuente de alimentación: 9 voltios.
- Rango de medición: 0 – 20 ppm.

- La sonda es de tipo polarográfico de alto rendimiento.
- Utiliza tapas de membrana de fluido electrolítico.
- Posee alta precisión y robustez.
- Puede compensar automáticamente la temperatura dentro del rango: 0 – 50 °C.



Figura 61. Medidor de oxígeno disuelto

Fuente: (Marine, 2018)

Además, se prevé los siguientes puntos para su utilización (Marine, 2018):

- La sonda se puede sumergir en el agua hasta una profundidad de 2,54 centímetros.
- El medidor no es impermeable y debe operar sobre un área limpia y seca.
- Para la toma de lecturas en el medidor, se debe esperar de 1 a 2 minutos para que se realice la compensación de temperatura.
- Para limpieza de la sonda, debe utilizar lana de acero pocas veces al año para evitar el vidriado.

Dentro del medidor para oxígeno disuelto consta de varios elementos como pantalla de visualización, amplificadores, conversor análogo – digital, entre otros, visto en la Figura 62. Además, se diagnostica la salida de voltaje variable para que se envíe ese dato al Arduino.



Figura 62. Diagrama interno del medidor para Oxígeno Disuelto

Fuente: Elaboración Propia

El medidor posee el amplificador HA17324A (Figura 63) que posee las siguientes características (Alldatasheet, 2018):

- Voltaje de alimentación máxima V_{CC} de 32 V.
- Corriente de alimentación de 50 mA.
- Temperatura de operación entre -20 a $+75$ °C.
- Cuatro salidas de voltaje: V_{out1} , V_{out2} , V_{out3} y V_{out4} .
- V_{EE} representa la tierra o negativo.

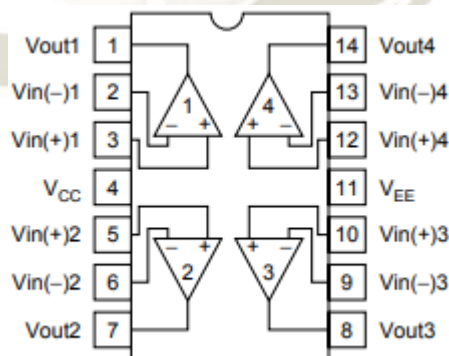


Figura 63. Circuito amplificador HA17324A

Fuente: (Alldatasheet, 2018)

Como la salida de voltaje varía entre 0 a 9 V por la batería que utiliza, se coloca una resistencia que servirá como caída de voltaje e ingrese a la medición A/D del Arduino Nano, visto en la Figura 64.

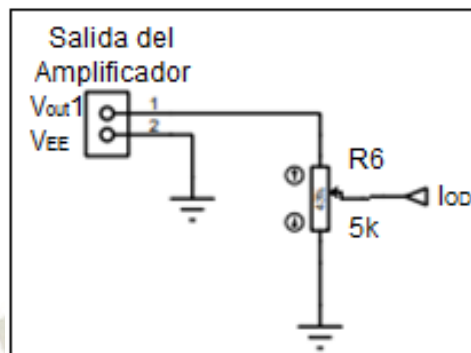


Figura 64. Diagrama circuital para la medición de oxígeno disuelto

Fuente: Elaboración Propia

Se asume una resistencia de precisión de 5 kΩ para determinar la corriente que entra a un pin A/D del Arduino Nano. Además, se estableció poner a la mitad del valor de resistencia para que su voltaje varía entre 0 a 4,5 V.

$$I_{OD} = \frac{9\text{ V}}{5\text{ k}\Omega} = 1.8\text{ mA}$$

3.3.4 Sensor de posicionamiento

Para ubicar la posición del vehículo de superficie no tripulado, se opta por escoger un sensor GPS (Figura 65) de la marca Ublox, el cual posee las siguientes características (Alldatasheet, 2018):



Figura 65. Sensor GPS Neo-6M-0-001

Fuente: (Naylamp, 2018)

- Posee una antena cerámica, tal como se indica en la Figura 57.

- Compatible con varios controladores de vuelo.
- Posee memoria EEPROM para guardar datos en caso de que el módulo se desconecte de manera imprevista.
- Tiene un indicador de señal, mediante el encendido de un led.
- Voltaje de alimentación: 3 – 5 Vdc.
- Velocidad de transmisión: 9600 bps (baudios por segundo).
- Tiene un interfaz serial UART 5V.

3.3.5 Sensor de imagen

Para la navegación del vehículo de superficie no tripulado, se opta por escoger un cámara CMOS (Figura 66) de la marca Raspberry Pi, el cual posee las siguientes características (Raspberrypi, 2018):

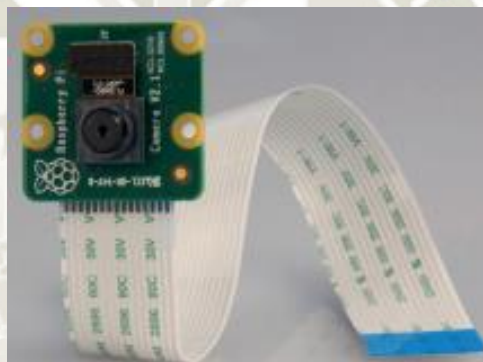


Figura 66. Cámara para Raspberry Pi

Fuente: (Raspberrypi, 2018)

- Alta definición y sensibilidad.
- Baja interferencia y captura de imágenes.
- Posee un conector CSI, visto en la Figura 66.
- Gracias al bus CSI, posee altas velocidades de transmisión de datos.
- El sensor posee un módulo de enfoque fijo.
- En la Tabla 22, se expone las especificaciones técnicas.

Tabla 22.

Características técnicas de la cámara V2

Características	Descripción
Tipo	Sony IMX 219 PQ CMOS
Módulo	Enfoque fijo
Resolución	8 megapíxeles
Resolución de imagen fija	3280 x 2464
Velocidad de transferencia de imagen máxima	1080p: 30fps (codificar y decodificar) 720p: 60fps
Pines de conexión	15
Rango de temperatura	- 20 a 60 °C
Funciones de control de imagen	Control automático de exposición.
	Balance de blancos automáticos.
	Filtro de banda automático.
	Detección de luminancia a 50/60 Hz.
	Calibración automática del nivel de negro.

Fuente: (Raspberrypi, 2018)

3.3.6 Sensor para medición de capacidad de líquido

Para que no sobrepase el límite en las muestras de agua, se opta por escoger un sensor de flujo YF-S201. Este instrumento se instala en la tubería por donde el agua circula a través del sensor y hace mover una turbina en forma de aspas que está unido a un imán que

activa el sensor de efecto hall (Figura 67), estas mediciones hacen que el sensor permanezca seco y seguro. Además, posee las siguientes características (Naylamp, 2018):



Figura 67. Sensor YF-S201

Fuente: (Naylamp, 2018)

- Medir caudal en tuberías de 1/2" de diámetro.
- Compatible con plataformas Atmega, Pic, Raspberry Pi y PLCs.
- Posee tres cables de conexión: cable rojo (Vcc = 5V), cable negro (tierra o negativo), y cable amarillo (salida de pulsos).
- Capacidad de carga: 10 mA.
- Factor de conversión: 7,5 (utilizado para el software).
- Precisión de +10% y temperatura de funcionamiento entre - 25 a 80 °C.
- El flujo de agua se determina en litros/minuto bajo la Ecuación 9.

$$\text{Flujo} = \frac{\text{Pulsaciones del sensor (hz)}}{\text{Factor de conversión}} \quad (9)$$

3.4 Selección de actuadores

El barco de superficie no tripulado posee dos servomotores que funcionan con 7,2 V y 700 mAh. Estos motores se utilizan ya que poseen un buen torque y pueda desplazarse sobre el agua. Un servomotor por lo general viene provisto de un motor de corriente continua más una caja de engranajes. Por último, posee dos cables de conexión, visto en la Figura 68.



Figura 68. Servomotor para el vehículo de superficie no tripulado

Fuente: (AliExpress, 2018)

Para la extracción del agua se utiliza un pequeño motor de corriente continua y una turbina en forma de aspas, visto en la Figura 69. Se ha seleccionado la misma fuente de alimentación que utilizan los servomotores del barco.



Figura 69. Mini bomba para succión del líquido

Fuente: Elaboración Propia

3.5 Selección de los elementos de control

3.5.1 Selección de plataformas para programación

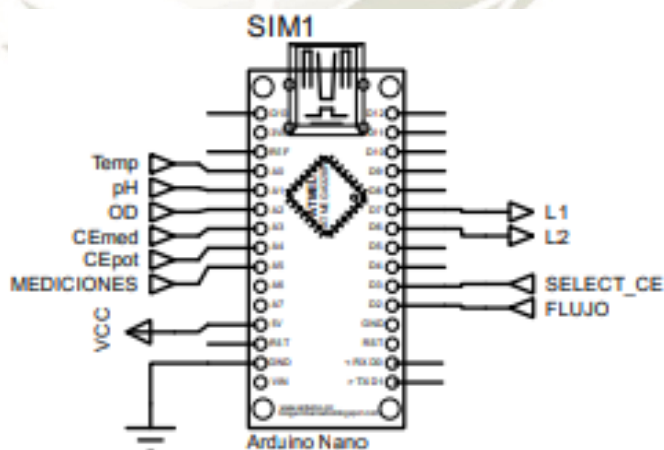


Figura 70. Arduino Nano

Fuente: Elaboración Propia

Se tiene varios sensores y actuadores para el desarrollo del sistema, por lo que se utiliza la plataforma Arduino Nano (Figura 70) para las siguientes funciones:

- Lectura de los sensores de temperatura, pH, conductividad eléctrica, oxígeno disuelto y flujo.
- Selección e indicación para la medición de conductividad eléctrica.

Además, se utiliza la Raspberry Pi3 B+ para las siguientes acciones, visto en la Figura 71.

- Conexión de la cámara.
- Colocación de una microSD, para instalación del programa Raspbian.
- Pines para el circuito de control (puente H) hacia los actuadores.
- Conexión serial del GPS.
- Conexión USB a mini USB para recepción de lectura de sensores mediante el Arduino Nano.

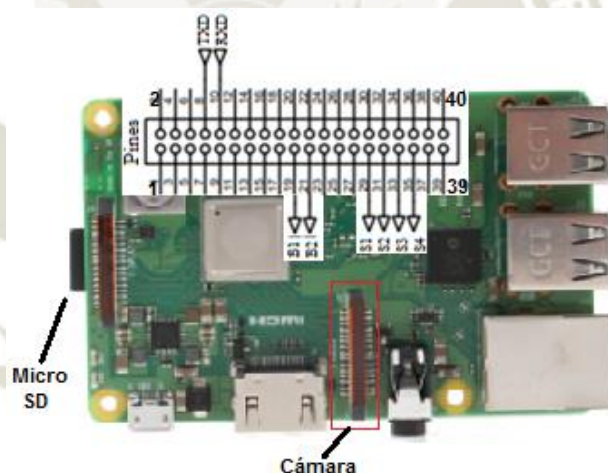


Figura 71. Raspberry Pi3

Fuente: Elaboración Propia

3.5.2 Selección para control de motores

Para los dos servomotores y un motor de corriente continua se colocan dos integrados SMD L298, visto en la Figura 72. La Raspberry Pi3 envía señales de control al puente H y este a su vez se conecta a los motores para efectuar el movimiento establecido.

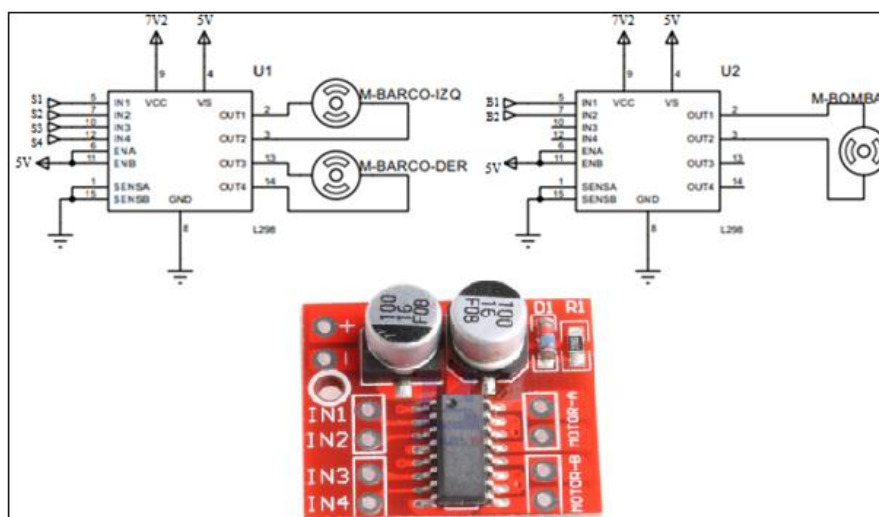


Figura 72. Puente H

Fuente: (Alldatasheet, 2018)

3.6 Selección de fuentes de alimentación

Se escoge una batería externa o power bank (Figura 73.a) para los circuitos que operan a 5 V, es decir los sensores, microcontrolador Arduino Nano y microprocesador Raspberry Pi3. Son baterías de polímero de litio (Li-Po) de 20.000 miliamperio - hora. En cambio, se seleccionó una batería de Ni-Cd (vino en el vehículo de superficie no tripulado) para la alimentación de los motores, visto en la Figura 73.b).

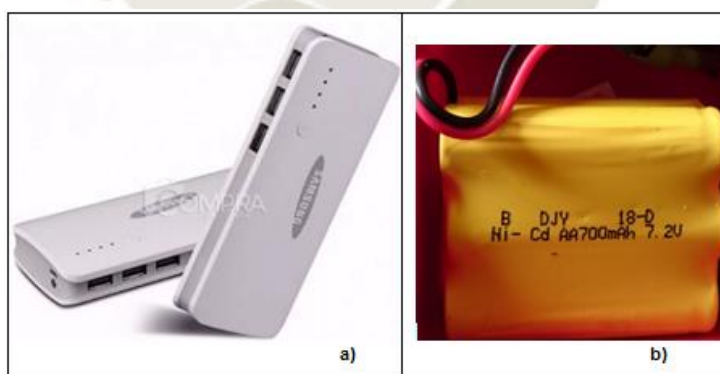


Figura 73. Batería Li-Po y Ni-Cd

Fuente: Elaboración Propia

3.7 Fuente de agua viables

Para determinar el tipo de fuente de agua adaptable al presente proyecto, se realiza una recopilación de los sensores seleccionados (apartados 3.3.1 al 3.3.3), visto en la Tabla 23.

Tabla 23.

Sensores seleccionados

Parámetros para calidad del agua		Sensores		
		Gaohou	VA 10200	Pinpoint II
Potencial de Hidrógeno	pH	0 – 14	-	-
Temperatura	°C	0 – 80	0 – 70	0 – 50
Conductividad eléctrica	uS/cm	-	0 – 1999	-
Oxígeno Disuelto	ppm=mg/l	-	-	0 – 20

Fuente: Elaboración Propia

En la Tabla 24 se colocan parámetros de calidad del agua investigados para los siguientes ríos: sumbay, blanco, chili y yura (Córdova, 2016)

Tabla 24.

Calidad del agua en algunos sectores de Perú en el año 2016

Parámetros para calidad del agua		Sectores (Ríos)				
		Sumbay	Blanco	Chili	Yura	Tingo
Potencial de Hidrógeno	pH	8.0	7.9	7.5	8.6	8.3
Temperatura	°C	15	16.7	15.5	15.2	15
Conductividad eléctrica	uS/cm	121	3190	221	733	3280
Oxígeno Disuelto	ppm=mg/l	7.25	7.99	7.29	8.31	11.31

Fuente: (Córdova, 2016)

Observando las Tablas 23 y 24, los medidores son accesibles para ciertos Ríos (Sumbay, Chili, Yura) debido al parámetro de conductividad eléctrica.

Capítulo IV

4. SOFTWARE DEL SISTEMA

4.1 Dinámica de fluidos computacional CFD

A través del software de simulación ANSYS 18 se realiza el análisis CFD, cuya función es determinar si los nuevos componentes del vehículo de superficie no tripulado causan inestabilidad cuando se encuentre en funcionamiento. Para realizar el análisis CFD se debe realizar los siguientes pasos:

- Se realiza la extracción de volumen en la pantalla Workbench del Ansys 18, tomando en cuenta que el formato debe ser “.step” para que el software se abra. En la Figura 74 se observa el modelo en la pantalla Space Claim de Ansys.

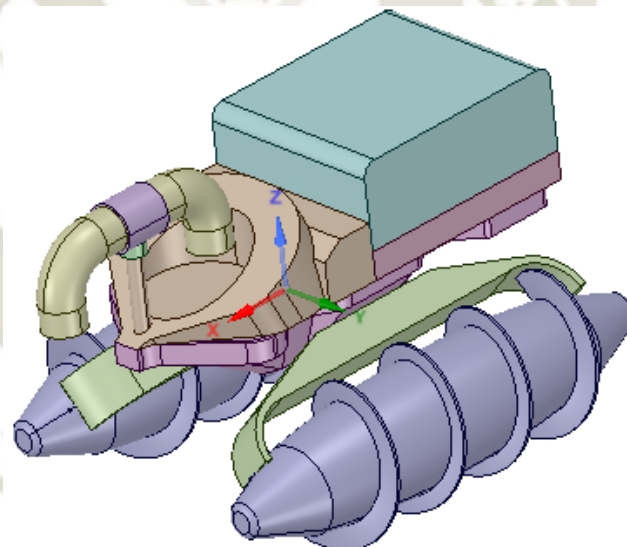


Figura 74. Modelo en la pantalla Space Claim

Fuente: Elaboración Propia

- Se coloca un enclosure para la simulación del viento en el barco, donde se ubican las entradas y salidas cuando se realice el mallado, visto en la Figura 75.

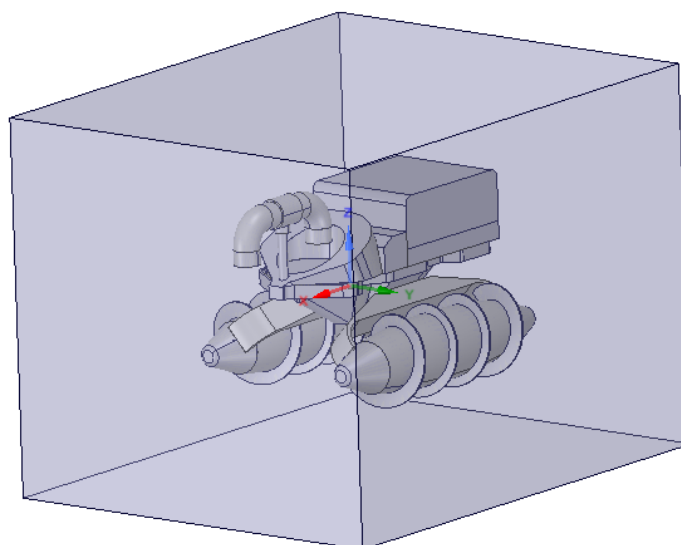


Figura 75. Vehículo junto con el enclosure

Fuente: Elaboración Propia

4.1.1 Parámetros de mallado

La calidad de malla Skewness debe encontrarse dentro del intervalo 0.22 a 0.25, para determinar buenos parámetros de mallado (Tabla 25), el mismo que es refinado con los siguientes parámetros de calidad y tamaño.

Tabla 25.

Parámetros de mallado

Parámetro de malla	Propiedad o medida
Relevance	0
Size function	Curvature
Method	Tetrahedrons
Physics preference	CFD
Smoothing	High
Solver preference	Fluent
Skewness (Average)	0,24

Fuente: Elaboración Propia

4.1.2 Entrada y salida de aire

La entrada de aire se coloca delante del bote ya que el vehículo de superficie no tripulado avanza frontalmente y la salida de aire es por la parte posterior, tal como se muestra en la Figura 76.

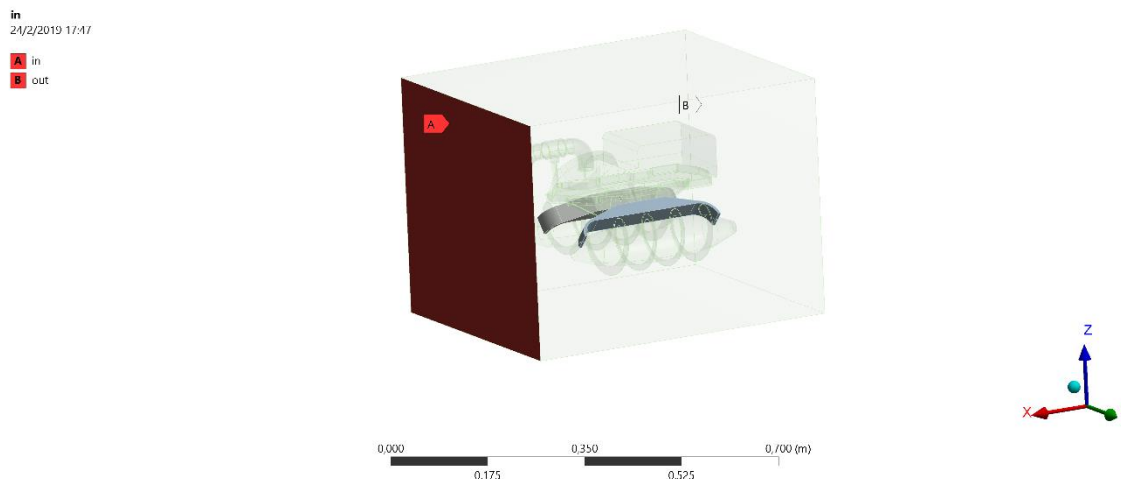


Figura 76. Entrada y salida de aire

Fuente: Elaboración Propia

4.1.3 Flujo turbulento

La resistencia del viento al avance del vehículo ocasiona un flujo turbulento. Los comandos ubicados para que en el enclosure realice un cálculo mediante flujo turbulento y remolinos en los cambios de sección se determinan a continuación:

- Model (Viscous)
- k-epsilon (2 egn)
- RNG
- Swirl Dominated Flow

En la ventana de Boundary Conditions se coloca el valor inicial del viento de 17 m/s donde se realiza el ensayo, visto en la Figura 77.

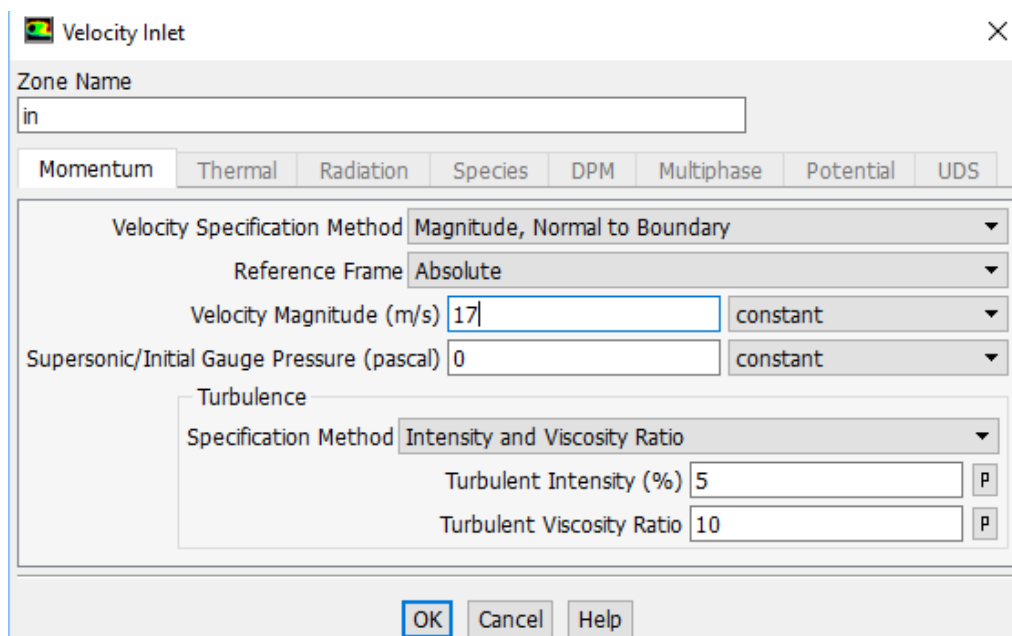


Figura 77. Workbench

Fuente: Elaboración Propia

A continuación, se detallan todos los procesos desarrollados en el programa de Arduino y Python para la Raspberry Pi3.

4.2 Programa en Arduino nano

Se desarrolla el software en el programa Arduino versión 1.8.1, donde se definen las librerías y variables para la medición de sensores, luego se configura los puertos de entrada/salida y se procede con la realización del programa, visto en la Figura 78.

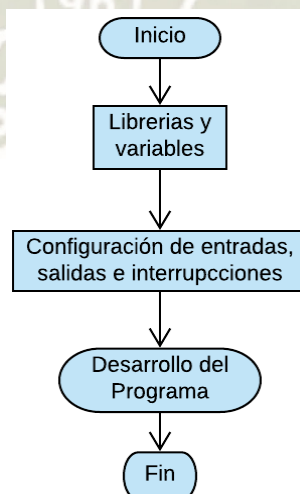


Figura 78. Bosquejo del programa en Arduino

Fuente: Elaboración Propia

Dentro del desarrollo del programa se tiene las siguientes condiciones:

- Programa para determinar el flujo del líquido.
- Se dispone de un interruptor (A5) para activar las mediciones de los sensores de parámetros de calidad del agua del Arduino a la Raspberry Pi3. Este paso es necesario, ya que el Arduino pasa constantemente conectado a la Raspberry Pi3 y sus pines desconectados mandar ruido (datos incoherentes).
- Programa para determinar las mediciones de temperatura, pH y oxígeno disuelto.
- Se dispone de un pulsador (interrupción D3) y leds para indicar el modo de operación para el sensor de conductividad eléctrica. Para luego determinar la medición de conductividad.
- Visualización de parámetros.

4.2.1 Diagrama para medición de temperatura

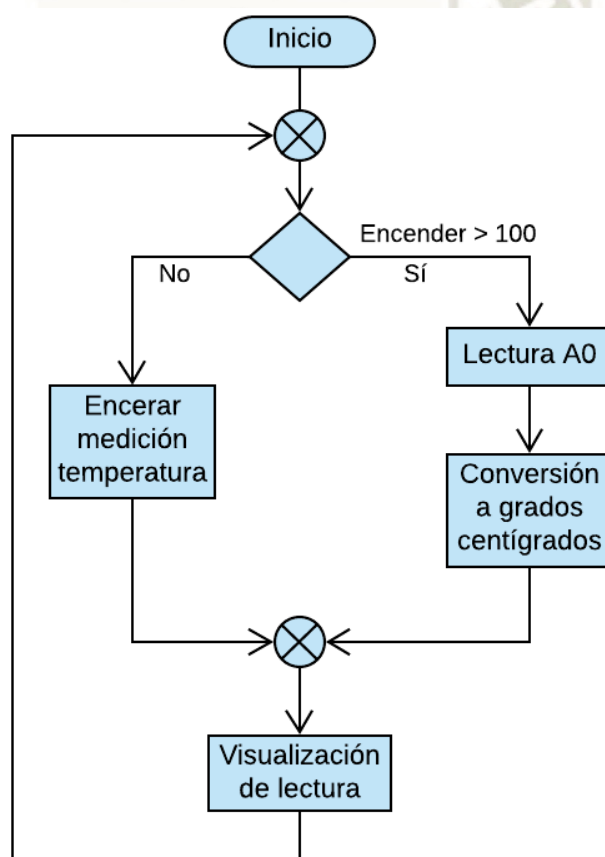


Figura 79. Medición de temperatura

Fuente: Elaboración Propia

En la Figura 79 indica cómo se realiza una medición en el pin análogo/digital A0, mediante la Ecuación 10 se calcula la temperatura en grados centígrados y se visualiza la lectura para que esta sea leída por la Raspberry Pi3.

$$T [^{\circ}\text{C}] = A0/37.7 \quad (10)$$

4.2.2 Diagrama para medición de pH

En la Figura 80 indica cómo se realiza mediciones en el pin análogo/digital A1, se realiza un promediado de 20 datos con un retraso de 50 milisegundos y mediante la Ecuación 11 se determina el valor de pH. Por último, se visualiza la lectura para que esta sea leída por la Raspberry Pi3.

$$\text{pH} = A1/145 \quad (11)$$

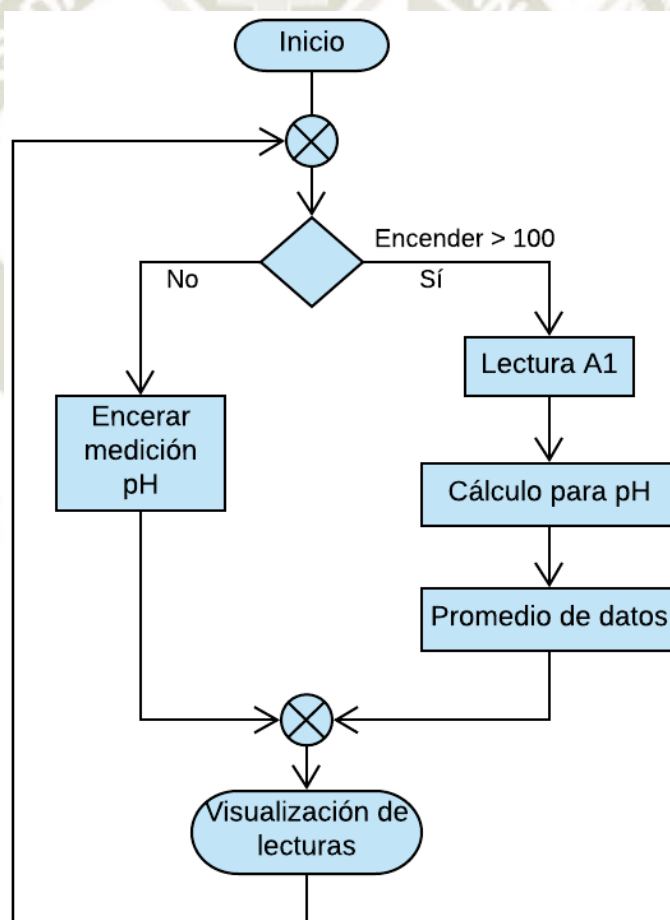


Figura 80. Medición de pH

Fuente: Elaboración Propia

4.2.3 Diagrama para medición de oxígeno disuelto

En la Figura 81 indica cómo se realiza mediciones en el pin análogo/digital A2, se realiza un promediado de 20 datos con un retraso de 50 milisegundos y mediante la Ecuación 12 (corrección de la Ecuación 17) se determina el valor de oxígeno disuelto. Además, se coloca una condición de si el medidor es menor a 0,1 se encere la medición debido a que el medidor manda valores negativos y eso no puede aceptar el Arduino. Por último, se visualiza la lectura para que esta sea leída por la Raspberry Pi3.

$$OD [ppm] = (0.121 * A2) - 56.3 \quad (12)$$

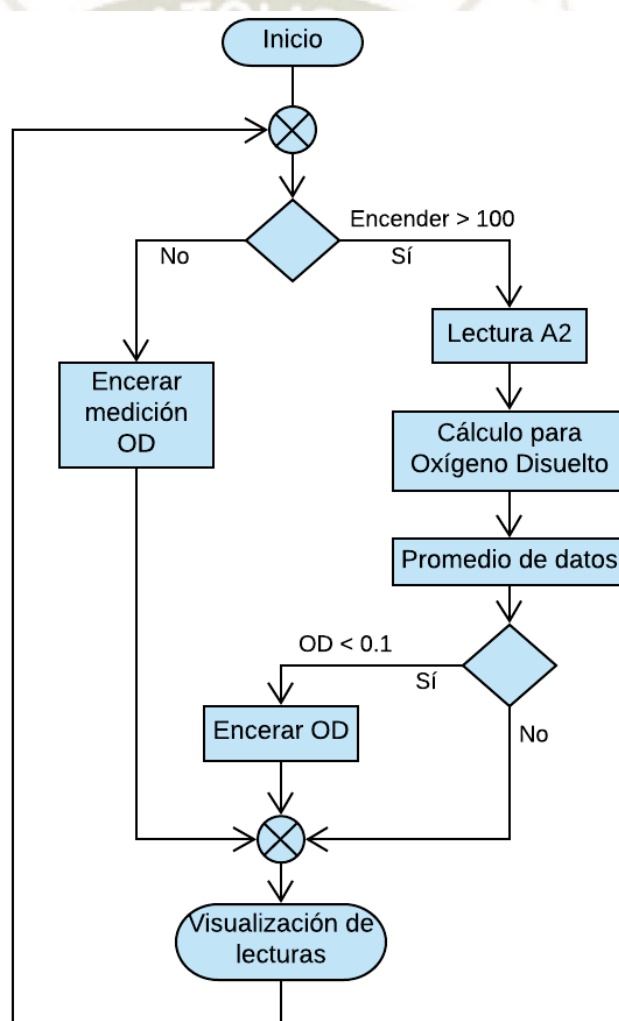


Figura 81. Medición de oxígeno disuelto

Fuente: Elaboración Propia

4.2.4 Diagrama para medición de conductividad eléctrica

Para las mediciones de conductividad eléctrica, el software se basó mediante la lógica de la Tabla 21. En la Figura 82 indica cómo se realizan condiciones, ingreso de ecuaciones, promediado de 20 datos de las ecuaciones (13 y 14) leídas por los canales A3 y A4; previo a estos pasos se debe activar la interrupción externa INT1 para que el pulsador vaya contando los modos de operación. Por último, se visualiza la lectura para que esta sea leída por la Raspberry Pi3.

Agua de uso residencial.

$$CE [uS/cm] = (0.8908 * A3) - 105.47 \quad (13)$$

Agua combinada con azúcar (95 y 5%).

$$CE [uS/cm] = (0.8395 * A3) + 53.754 \quad (14)$$

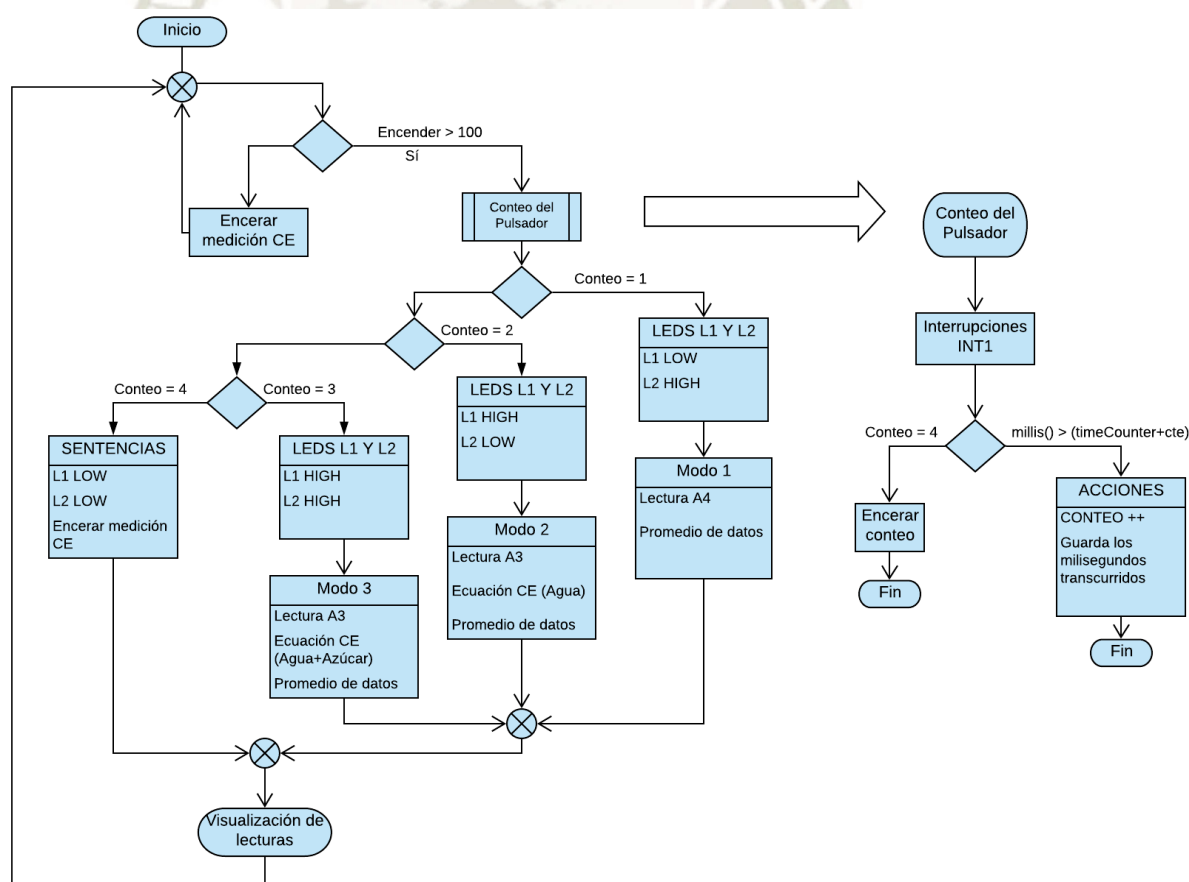


Figura 82. Medición de conductividad eléctrica

Fuente: Elaboración Propia

4.2.5 Diagrama para medición de Flujo

Para determinar el flujo del líquido se procede a utilizar la interrupción externa INT0. En la Figura 83 indica cómo se realiza el cálculo para determinar la frecuencia (Ecuación 15) y a su vez el flujo (Ecuación 9) medido en litros por minuto. Por último, se visualiza la lectura para que esta sea leída por la Raspberry Pi3.

Frecuencia.

$$\text{Frecuencia [Hz]} = (\text{Contador} * 1000) / 20 \quad (15)$$

Flujo.

$$\text{Flujo} \left[\frac{\text{litros}}{\text{min}} \right] = \frac{\text{Pulsaciones del sensor (Hz)}}{\text{Factor de conversión}}$$

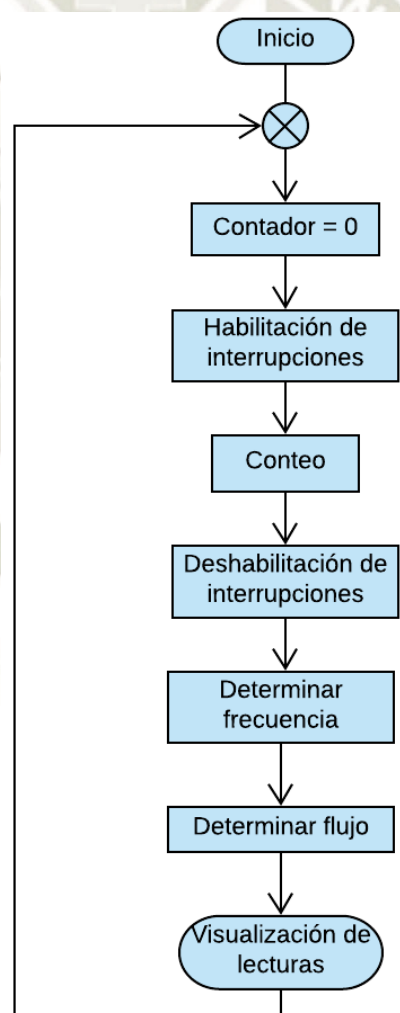


Figura 83. Medición de flujo

Fuente: Elaboración Propia

4.3 Programa en raspberry pi3 b+

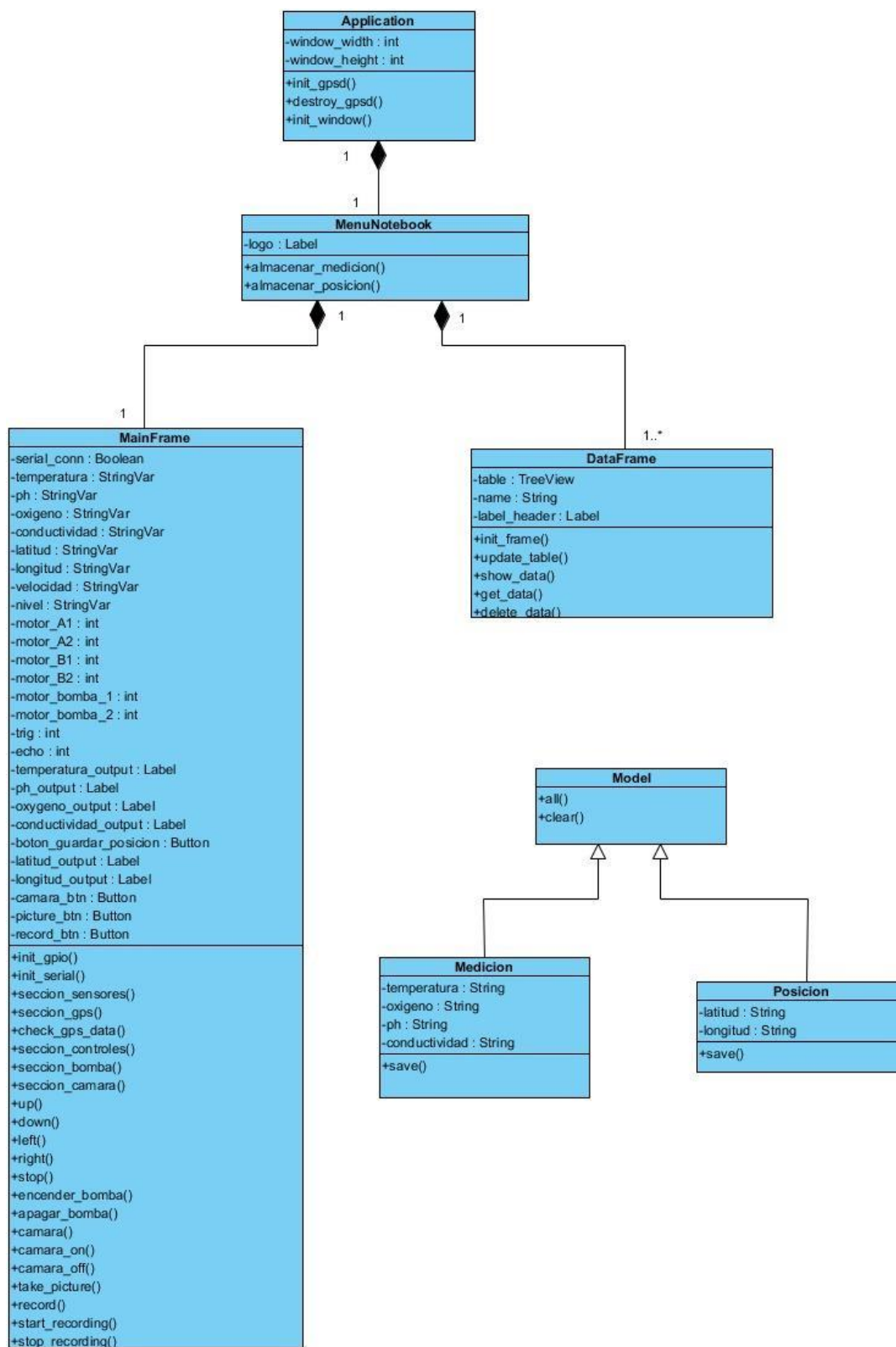


Figura 84. Estructura de la aplicación

Fuente: Elaboración Propia

Se desarrolla el software en la Raspberry Pi3 B+ instalando la aplicación raspbian y Python. Para empezar con el software se deben instalar todas las librerías para la cámara, GPS, variables, extensiones (buffer), entre otros, los cuales servirán al desarrollo de la aplicación, luego similar a lo que está indicado en la Figura 78 (del Arduino) se procede a definir las variables del programa, tal como se indica en la Figura 84.

Luego se procede a desarrollar cada parte del programa, indicado a continuación:

4.3.1 Recepción de lectura de sensores en la Raspberry Pi

Para establecer una buena recepción en la Raspberry Pi de los parámetros enviados desde el Arduino Nano, se debe realizar los siguientes pasos (Figura 85):

- El programa verifica la conexión entre el Arduino y la Raspberry Pi. En caso de no establecer la conexión, mostrará un mensaje de alerta.
- Una vez establecido la conexión, se procede a leer y visualizar los parámetros.
- Por último, se tiene un botón para almacenar los parámetros en un buffer (extensión) que servirá para la base de datos.

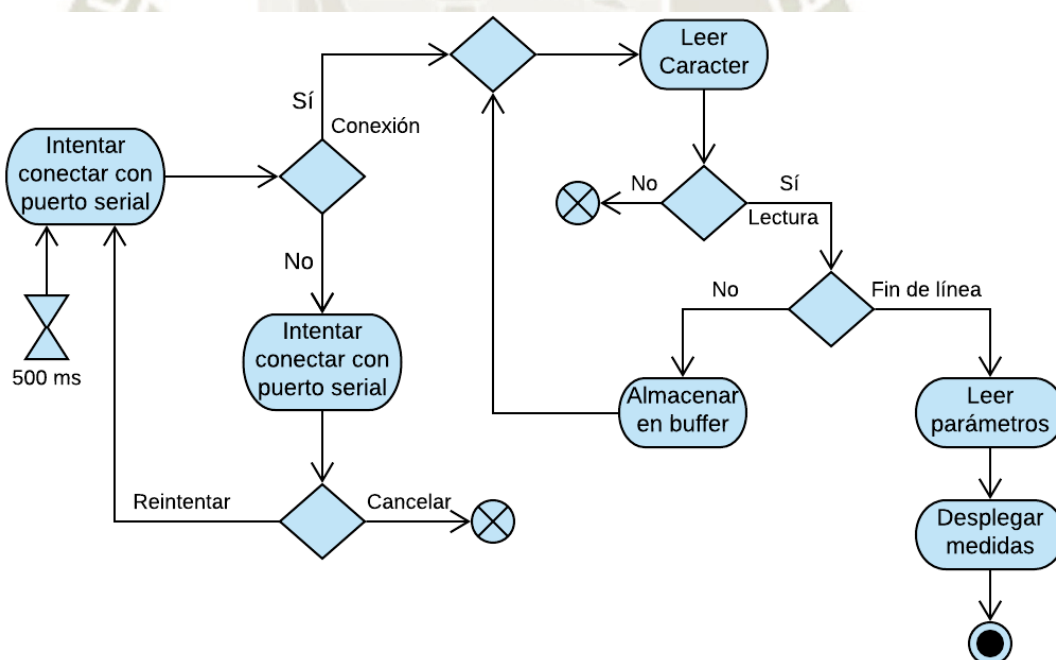


Figura 85. Lectura de sensores en la Raspberry Pi3

Fuente: Elaboración Propia

4.3.2 Modelo de datos

En la Figura 86 se observa la estructura de una base de datos para la captura de los datos medidos en campo, es decir, guardar los parámetros de temperatura, pH, oxígeno disuelto, conductividad eléctrica y fecha para los parámetros de calidad del agua. Además, guarda parámetros de la posición (latitud y longitud) del vehículo de superficie no tripulado. Se realizó un sistema de gestión de base de datos con SQLite, el cual es compatible con Python.

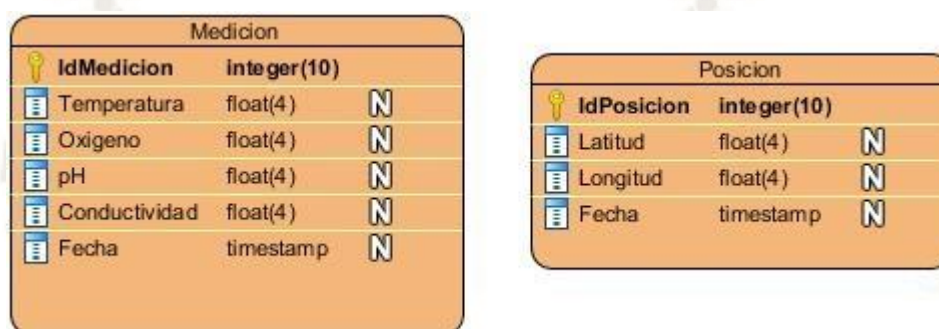


Figura 86. Modelo de datos
Fuente: Elaboración Propia

4.3.3 Programa para activación/desactivación de motores

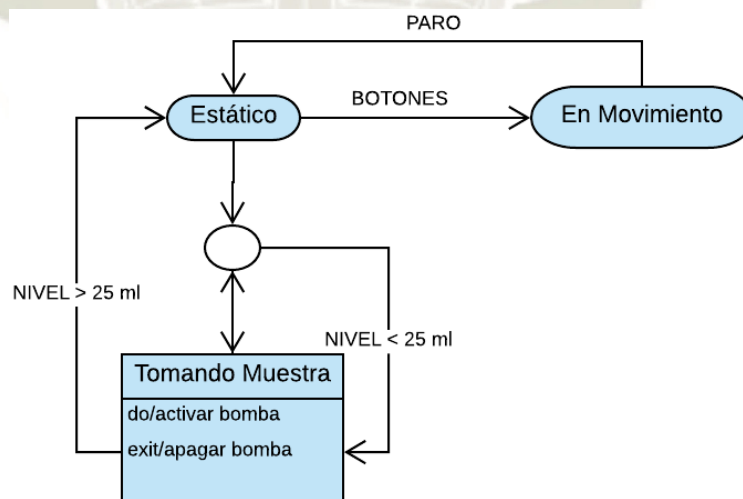


Figura 87. Estados del sistema
Fuente: Elaboración Propia

La activación y desactivación de los motores tanto para el vehículo de superficie no tripulado como la bomba que extraerá el líquido se realiza a través de botones vistos en la aplicación Python. Mediante el uso de botones se puede desplazar el vehículo, además posee un botón adicional de paro que indicará la desactivación o apagado de los motores. También se tiene otros botones para el activado/desactivado del motor de la bomba para la extracción del líquido, visto en la Figura 87.

Para el uso de la bomba aparte de los botones, se tiene una función la cual almacena el volumen para que no se riegue el líquido del envase, tal como se indica en la Figura 88. Entonces tiene dos formas de parar la bomba, la primera es mediante el botón de apagado y la segunda es con una lógica interna indicando que no debe sobrepasar los 25 mililitros de capacidad.

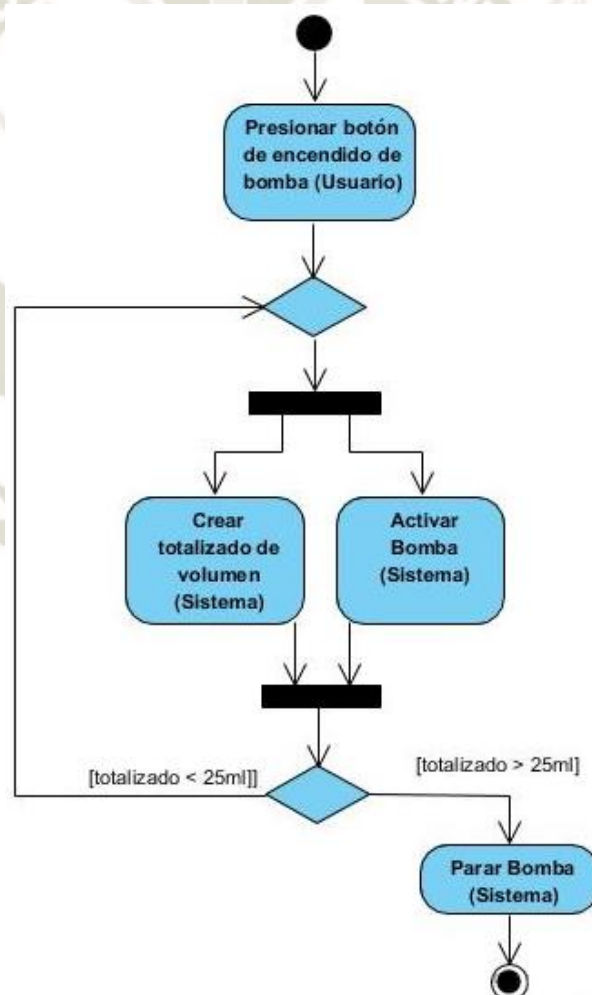


Figura 88. Toma de muestras

Fuente: Elaboración Propia

4.3.4 Lectura del GPS

Los pasos que se deben realizar para la activación del GPS (Figura 89) en la Raspberry son:

- Enmascarar el servicio para que cambie de grupo el archivo y luego poder habilitar los permisos de lectura en el puerto: /dev/ttyAMA0
- Activar el Daemon con GPSD (sirve para leer del puerto serial).
- Utilizar la librería en el programa GPS3. Esta librería se comunica con Daemon mediante comunicación TCP.
- Leer datos de la librería (Latitud, Longitud, Velocidad).
- Se imprime en el programa los datos.

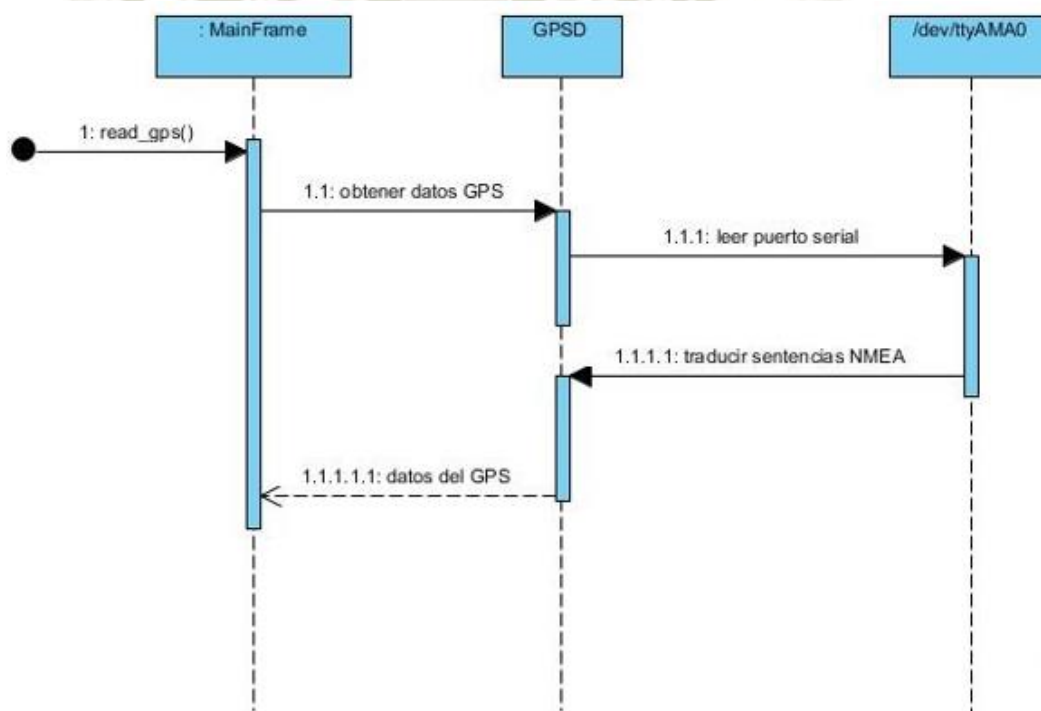


Figura 89. Lectura del GPS

Fuente: Elaboración Propia

4.4 Visualización del sistema en Python

En la Figura 90, se observa el sistema desarrollado en la aplicación Python, que es producto de todo el software desarrollado en el apartado anterior. Adicional se tiene

funciones para la cámara y son: video en tiempo real, toma fotos y guarda videos del recorrido establecido.

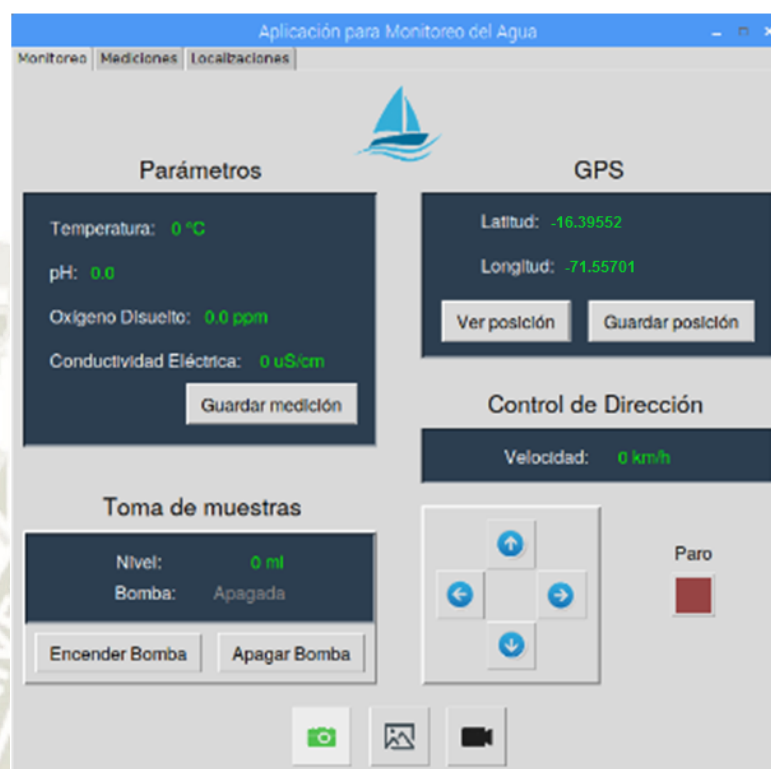


Figura 90. Visualización de la aplicación

Fuente: Elaboración Propia

4.5 Comunicación inalámbrica

Para que la visualización de la aplicación Python con el vehículo de superficie no tripulado sea a través de una comunicación inalámbrica, se dispone de sistema gráfico compartido con el computador, llamado VNC Viewer, visto en la Figura 91. Para tener la aplicación en el computador se debe tener la dirección IP del router, visto en la Figura 92 (Raspberrypi, 2018).

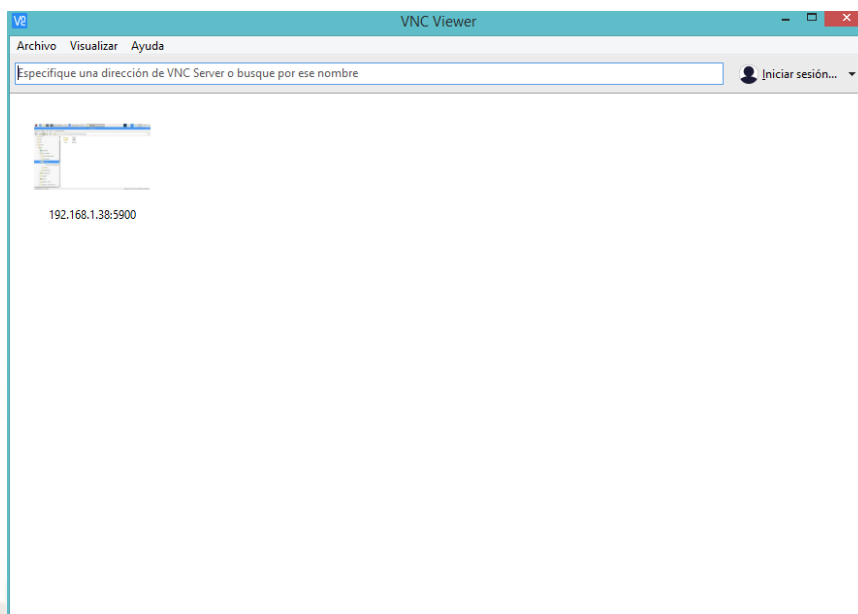


Figura 91. VNC Viewer

Fuente: Elaboración Propia

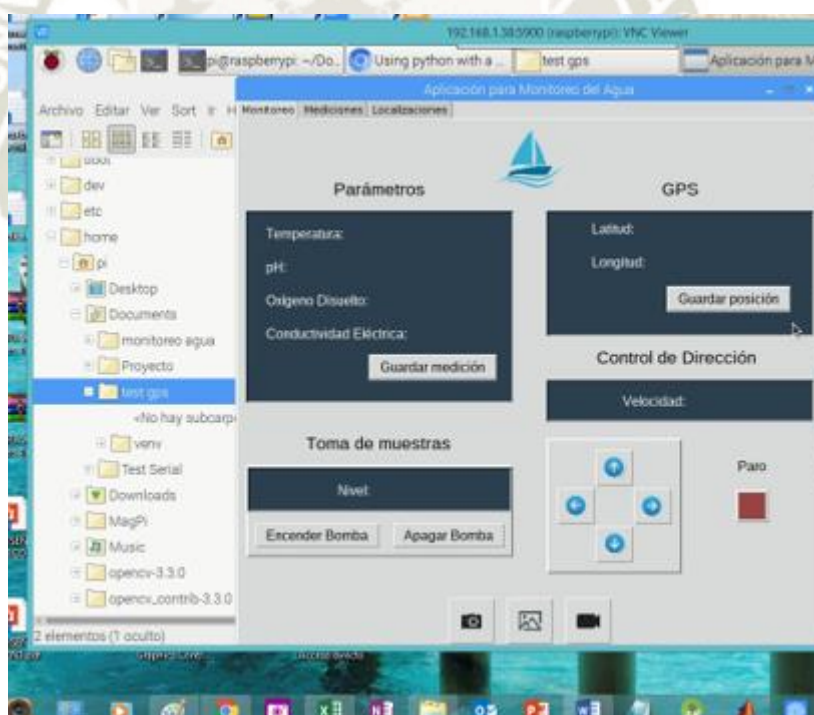


Figura 92. Comunicación inalámbrica entre el Computador y la Raspberry Pi

Fuente: Elaboración Propia

Capítulo V

5. PRUEBAS Y RESULTADOS

Las pruebas que se realizaron en el presente proyecto se detallan a continuación:

- Pruebas para determinar la velocidad y desplazamiento del vehículo de superficie no tripulado.
- Análisis CFD para el vehículo de superficie no tripulado
- Pruebas con los sensores de temperatura, pH, conductividad eléctrica y oxígeno disuelto.
- Pantallas de visualización con la aplicación Python.
- Comparación entre parámetros de la toma de muestras realizadas con los valores permitidos por el ECA.

5.1 Velocidad del vehículo de superficie no tripulado

Para determinar la velocidad del vehículo, se debe determinar el diámetro del eje del servomotor (Figura 93) y luego mediante un tacómetro determinar la rapidez expresado en revoluciones por minuto, visto en la Figura 94.

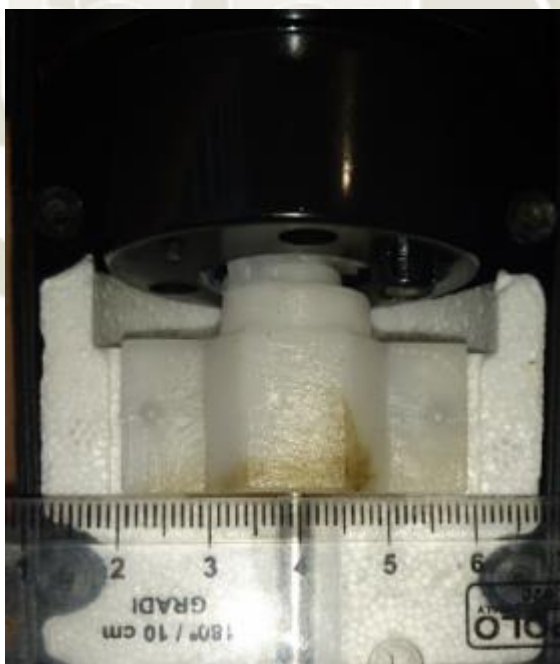


Figura 93. Eje del servomotor

Fuente: Elaboración Propia



Figura 94. Velocidad vista con un tacómetro digital

Fuente: Elaboración Propia

De las tres mediciones se escoge el promedio que es de 574.2 rpm (8.66 km/h) con el vehículo sin peso en movimiento. Sin embargo, cuando el vehículo de superficie no tripulado está situado sobre el agua con los elementos electrónicos instalados, se tiene una velocidad no mayor a 0.191 km/h, visto en la Figura 95 (datos obtenidos de velocidad del GPS).

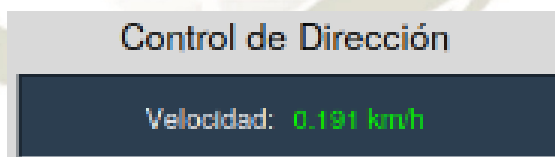


Figura 95. Velocidad con dato obtenido del GPS

Fuente: Elaboración Propia

5.2 Desplazamiento del vehículo de superficie no tripulado

Los datos de latitud y longitud del GPS vienen dados por el Sistema Geodésico Mundial WGS-84 que es un sistema de coordenadas, estos determinarán la posición del vehículo, visto en la Figura 96.

Localizaciones almacenadas			
No.	Latitud	Longitud	Fecha y Hora
1	-16.39552	-71.55701	2019-03-01 15:57:11.60637
2	-16.39551	-71.55702	2019-03-01 15:57:21.41071
3	-16.39551	-71.557	2019-03-01 15:57:29.59859
4	-16.39552	-71.55695	2019-03-01 15:57:56.78263
5	-16.39552	-71.55694	2019-03-01 15:58:03.74816
6	-16.39553	-71.55694	2019-03-01 15:58:11.37124
7	-16.39552	-71.55693	2019-03-01 15:58:14.20770
8	-16.39551	-71.55694	2019-03-01 15:58:17.72973
9	-16.39549	-71.55696	2019-03-01 15:58:23.60744
10	-16.3955	-71.55696	2019-03-01 15:58:29.60251

Figura 96. Posición WGS-84 obtenido del GPS

Fuente: Elaboración Propia

Como se visualiza en la Figura 96, el seguimiento de puntos es demasiado pequeño para ser vistos directamente en un archivo CAD o en el Google Maps; por ello se dispone a utilizar la Ecuación 16, donde se debe calcular la distancia entre dos puntos y multiplicado por una constante (TuTiempo, 2019).

$$\text{distancia [km]} = 100 * \sqrt{(\text{Lat}_2 - \text{Lat}_1)^2 + (\text{Long}_2 - \text{Long}_1)^2} \quad (16)$$

En la Tabla 26, se observa el desarrollo de la Ecuación 16 para el cálculo de la distancia en kilómetros y metros.

Tabla 26.

Cálculo de la distancia

Puntos	Latitud	Longitud	$(\text{Lat}_2 - \text{Lat}_1)^2$	$(\text{Long}_2 - \text{Long}_1)^2$	Km	Metros
1	-16,39552	-71,55701	1E-10	1E-10	0,00141	1,414
2	-16,39551	-71,55702	0	4E-10	0,002	2
3	-16,39551	-71,557	1E-10	2,5E-09	0,0051	5,099
4	-16,39552	-71,55695	0	1E-10	0,001	1

5	-16,39552	-71,55694	1E-10	0	0,001	1
6	-16,39553	-71,55694	1E-10	1E-10	0,00141	1,414
7	-16,39552	-71,55693	1E-10	1E-10	0,00141	1,414
8	-16,39551	-71,55694	4E-10	4E-10	0,00283	2,828
9	-16,39549	-71,55696	1E-10	0	0,001	1
10	-16,39550	-71,55696				
TOTAL				0.1717	17.17	

Fuente: Elaboración Propia

5.3 Pruebas y resultados del análisis CFD

Para la visualización de los resultados se colocó un streamline en el plano XZ, en donde se indica la velocidad y presión máxima, los cuales son parámetros que pueden influir cuando el vehículo de superficie no tripulado se encuentra en movimiento.

5.3.1 Velocidad

Como se observa en la Figura 97, la velocidad máxima de 2.41 m/s (sin peso) se encuentra en los motores que accionan el tornillo sin fin para la movilización del barco. No se muestran lugares con gran oposición al paso del aire, mostrando que es adecuado la distribución de los componentes electrónicos y mecánicos.

Para la verificación de las zonas de más incidencia se creó un plano en la dirección XZ que muestra las partes del barco que mayor oposición presentan. En la Figura 98 se observan las zonas del barco.

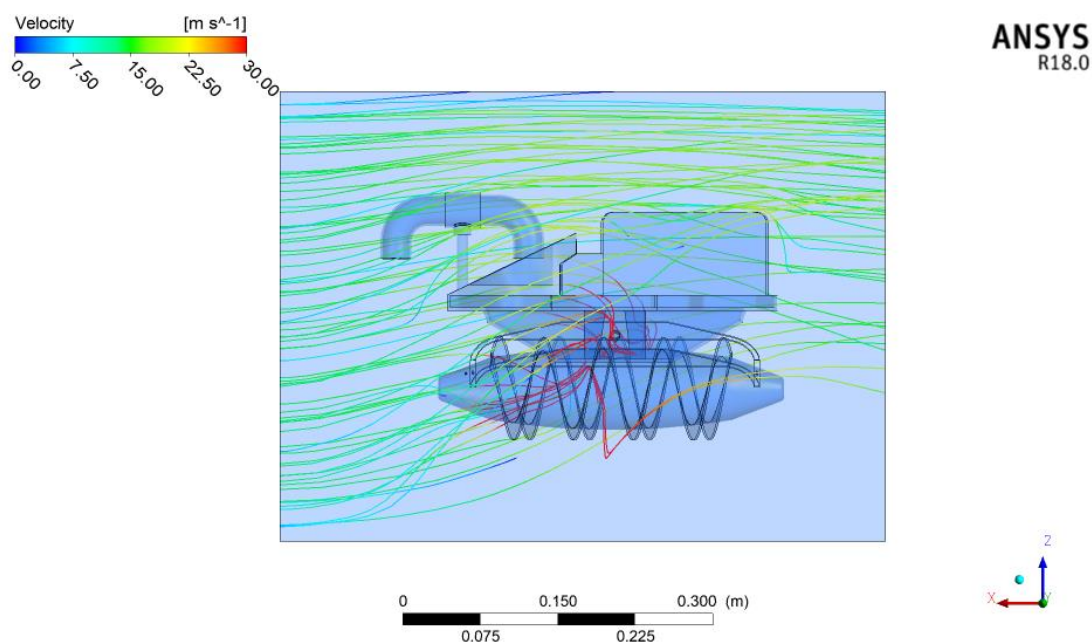


Figura 97. Velocidad
Fuente: Elaboración Propia

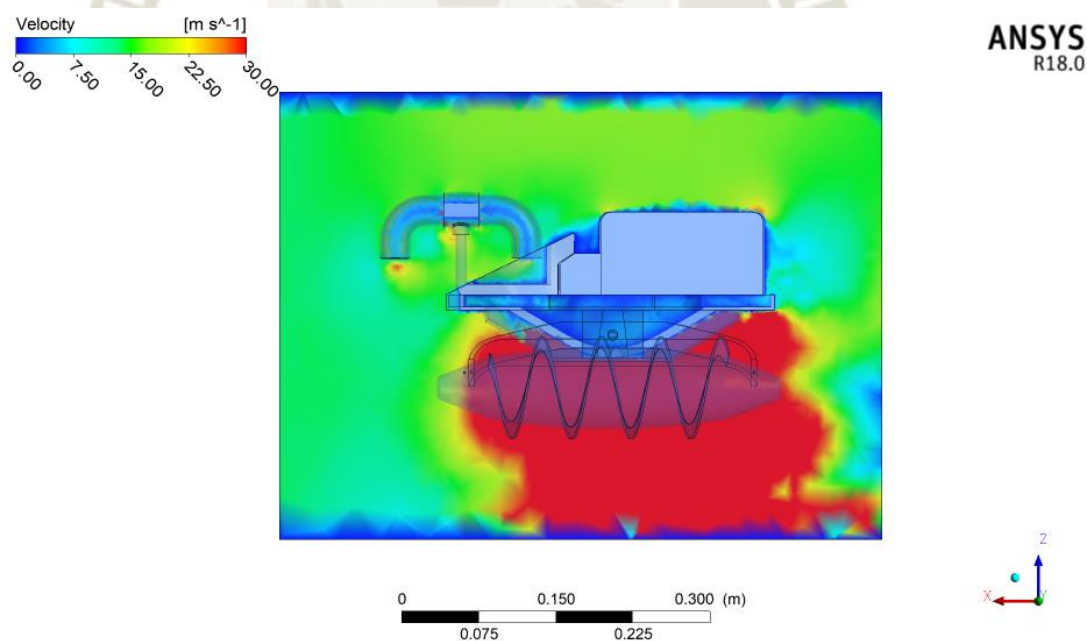


Figura 98. Zonas de incidencia con la velocidad
Fuente: Elaboración Propia

5.3.2 Presión

La presión máxima que soporta el barco se encuentra en la parte inferior del bote. En cambio, se tiene que la presión superior es la adecuada y no eleva al bote cuando está en funcionamiento.

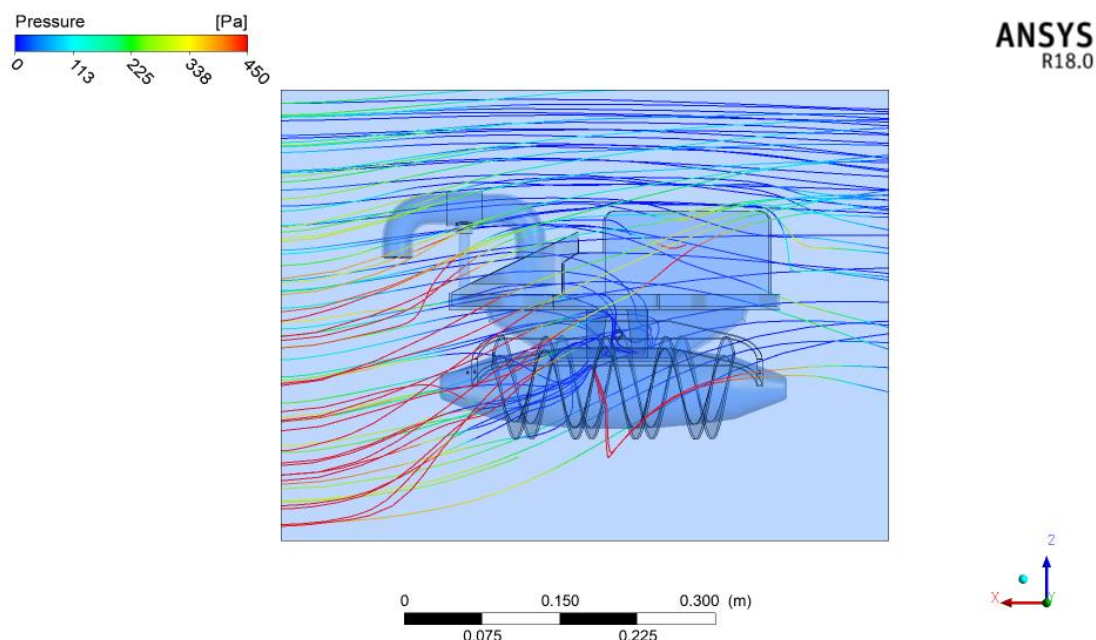


Figura 99. Presión

Fuente: Elaboración Propia

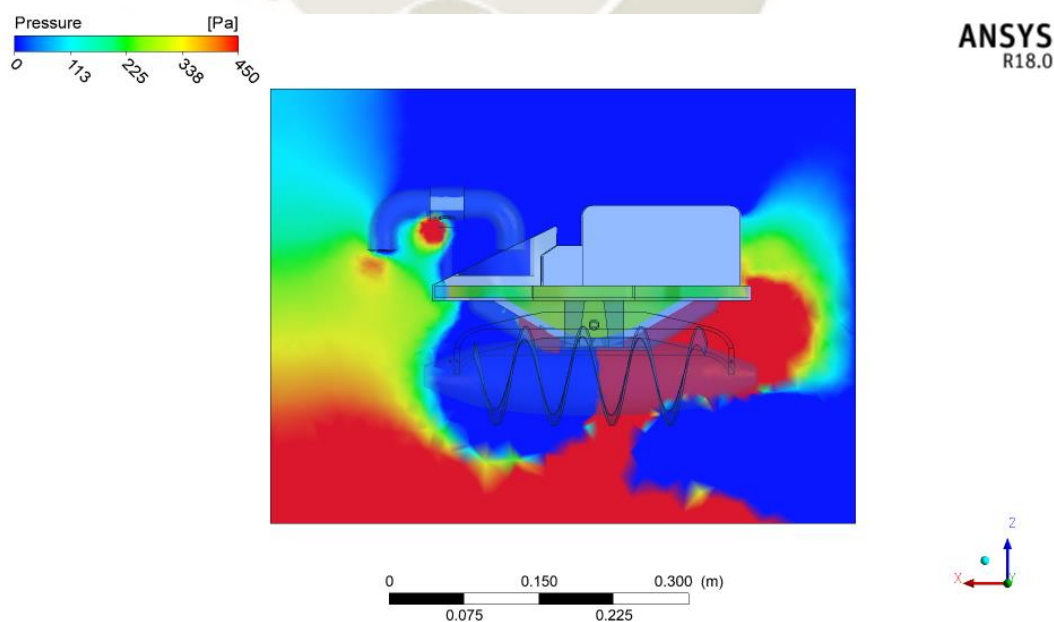


Figura 100. Zona de incidencia con la presión

Fuente: Elaboración Propia

En la Figura 100 se observa que el área del primer codo donde se acoplará el sistema de bombeo posee mayor presión, lo que podría causar que el barco se eleve, sin embargo, al colocar la manguera y como se colocó un viento moderado, en funcionamiento el barco no evidencia problema.

5.4 Pruebas con los sensores para medición de parámetros de calidad del agua

5.4.1 Pruebas de pH y temperatura

Las mediciones realizadas con agua de uso residencial son analizadas con el sensor comercial Gaohou ph0-14, visto en la Tabla 27 y Figura 101.

Tabla 27.

Condiciones iniciales de pH y temperatura

Análisis	Voltaje V	ADC	Factor (ADC/sensor)	Medición
pH	3.77	770	110	7.0
	3.78	773	110	7.0
	3.77	770	110	7.0
Temperatura	3.61	739	37	20 °C
	3.60	737	37	20 °C
	3.58	732	37	20 °C

Fuente: Elaboración Propia

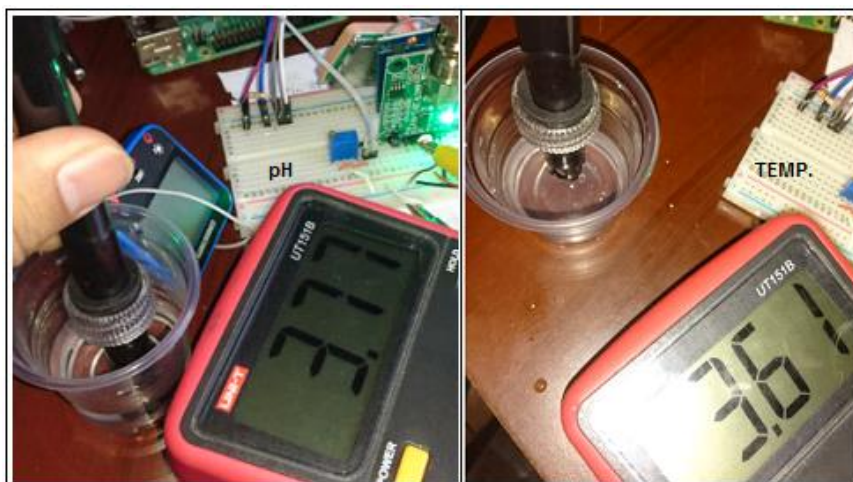


Figura 101. Condiciones iniciales de pH y temperatura

Fuente: Elaboración Propia

5.4.2 Pruebas de oxígeno disuelto

Como se observó en el apartado 3.3.3, para la toma de lecturas se debe esperar entre 1 a 2 minutos; por ello se ha realizado pruebas con agua de uso residencial y las mediciones se observan en la Tabla 28 y Figura 102.

Tabla 28.

Pruebas con el sensor de oxígeno disuelto

Voltaje V	ADC	ppm
2.23	456	0
2.32	475	2.2
2.58	528	8.6
2.62	536	9.6
3.68	753	35.9

Fuente: Elaboración Propia

Mediante la Tabla 28 se determina la ecuación con la cual se ingresará por software a la plataforma Arduino.

$$OD = 0.121 * AD - 55.198$$

(17)



Figura 102. Pruebas con el sensor de oxígeno disuelto

Fuente: Elaboración Propia

5.4.3 Pruebas de conductividad eléctrica

Este sensor posee ciertas limitaciones ya que no se puede probar con cualquier tipo de mezcla. A continuación, se establecen las siguientes ecuaciones cuando el sensor mide la presencia de agua para uso residencial y agua con azúcar (baja proporción).

Tabla 29.

Pruebas con el sensor de conductividad eléctrica

Voltaje V	ADC	uS/cm	Mezcla
1.50	307	168	Agua para uso
1.44	295	157	

1.49	305	166	residencial
1.45	297	148	Agua + Azúcar
1.48	303	178	

Fuente: Elaboración Propia

Mediante la Tabla 29 se determinan las Ecuaciones 13 y 14 con las cuales se ingresarán por software a la plataforma Arduino.



Figura 103. Pruebas del sensor CE para agua de uso residencial

Fuente: Elaboración Propia



Figura 104. Pruebas del sensor CE para agua con azúcar

Fuente: Elaboración Propia

5.5 Pantallas de visualización con la aplicación Python

En la Figura 90 (apartado 4.4), se observa que la pantalla principal consta de los siguientes elementos:

- Visualización de parámetros para la calidad del agua y un botón para guardar las lecturas tomadas, estas variables se almacenan en una viñeta llamada mediciones, visto en la Figura 105.

Mediciones realizadas					
No.	Temperatura	Oxígeno Disuelto	pH	Conductividad	Fecha y Hora
1	19.0	0.0	7.0	113.0	2019-02-27 16:48:02.1
2	20.0	1.2	7.2	123.0	2019-02-27 16:48:55.8
3	20.0	2.6	7.2	123.0	2019-02-27 16:51:08.0
4	19.0	1.6	7.2	132.0	2019-02-27 16:52:48.4

Figura 105. Parámetros de calidad del agua en Python

Fuente: Elaboración Propia

- Visualización de latitud, longitud y velocidad tomadas del sensor GPS, vistos en las Figuras 95 y 96. Adicional tiene un botón para ver la posición actual y se abre en una pantalla del Google Maps, visto en la Figura 106.

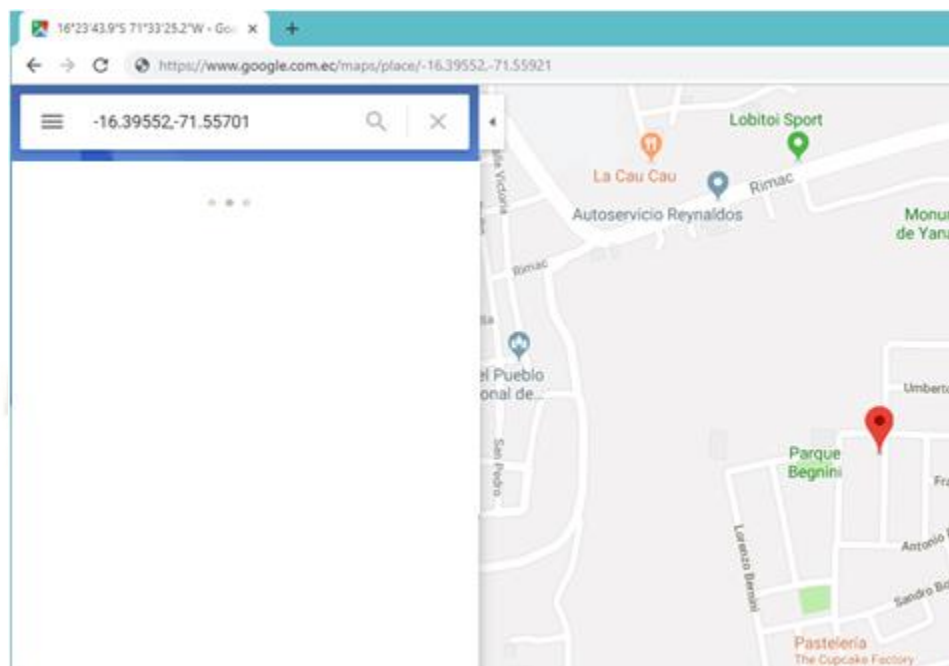
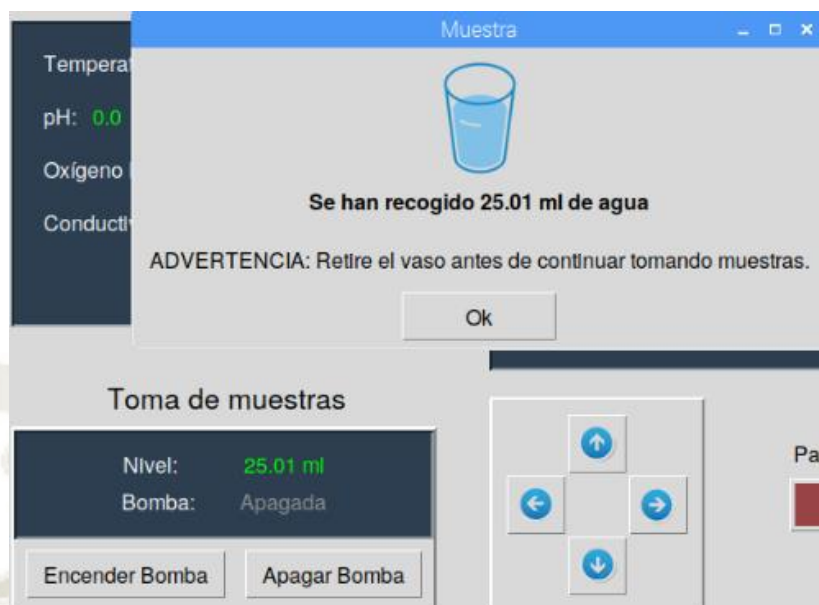


Figura 106. Visualización en Google Maps

Fuente: Elaboración Propia

- Activación/desactivación de la bomba para la extracción de agua. Además, dispone de la cantidad de mililitros de agua extraída; en caso de olvidarse durante el apagado, tiene un cierre automático cuando sobrepase los 25 ml, visto en la Figura 107.
- Posee tres botones que representan la activación/desactivación de la cámara, fotografía y video en tiempo real.
- Tiene cinco botones para el control de servomotores del barco, cuatro de ellos indica los movimientos respectivos (delantero, trasero, izquierda, derecha) y uno representa el paro a los dos servomotores.


Figura 107. Toma de muestras de líquido

Fuente: Elaboración Propia

5.6 Resultados de la muestra de agua para uso residencial

Comparando los datos de la Figura 105 (diferencia de temperaturas y valores promedios del pH, conductividad eléctrica y oxígeno disuelto) y Tabla 2 (Norma ECA), se evalúa los cuatro parámetros estudiados de calidad del agua, visto en la Tabla 30.

Tabla 30.

Resultados de la muestra de agua para uso residencial

Parámetros para calidad del agua		Pruebas Fig105	Norma ECA	Observaciones
Potencial de Hidrógeno	pH	7.2	6.5 – 9	Aceptable
Temperatura	°C	$\Delta 1$	$\Delta 3$	Aceptable
Conductividad eléctrica	uS/cm	123	1000	Aceptable
Oxígeno Disuelto	ppm=mg/l	1.8	≥ 5	No aceptable

Fuente: Elaboración Propia

5.7 Parámetros de la Calidad del agua medidos en algunos sectores de Arequipa

Parámetros para calidad del agua		Lagunas, Lagos y Manantiales			
		Tingo	Mejía	Yumina	Sabandía
Años		2019	2019	2019	2019
Potencial de Hidrógeno	pH	8.3	7.5	6.4	6.8
Temperatura	°C	15	22	16	18
Conductividad eléctrica	uS/cm	3280	5825	641	683
Oxígeno Disuelto	ppm=mg/l	11.31	3.6	5.24	4.72

Fuente: Elaboración Propia

5.8 Coordenada WGS84 tomados de sectores en Arequipa

Lagunas, Lagos y Manantiales	Coordenadas WGS84	
	Latitud	Longitud
Lago de Tingo	-16.43319	-71.56334
Laguna de Mejía	-17.14681	-71.86292
Manantial de Yumina	-16.4441	-71.48364
Lago de Sabandía	-16.4561	-71.4953

Fuente: Elaboración Propia

CONCLUSIONES

- Se logró construir un vehículo de superficie no tripulado, donde el sistema mecánico está compuesto por la embarcación y mecanismos. El vehículo fue diseñado tomando en consideración factores como resistencia de avance y el peso de las partes mecánicas y electrónicas. Además, se realizó una simulación aerodinámica para garantizar la flotabilidad del vehículo.
- Se desarrolló y observó mediante el programa Ansys los puntos de presión críticos que tuvo el vehículo de superficie no tripulado, dando como resultado que el mayor contacto se da en las hélices con respecto al agua y se estableció una velocidad de viento moderada; determinando que el vehículo no tuvo inestabilidad y además flotó de forma idónea sobre el agua (parte práctica).
- Se utilizó una cámara y un sensor GPS para determinar la trayectoria y velocidad del vehículo, este sensor muestra parámetros de posición WGS84 y permite poder calcular la distancia recorrida. Además, las trayectorias por donde puede desplazarse el vehículo de superficie no tripulado no están limitadas, ya que puede realizar los cuatro movimientos: hacia adelante, hacia atrás, los giros derecha e izquierda son de tracción diferencial. Por último, se controló el desplazamiento del vehículo mediante la aplicación VNC Viewer de forma inalámbrica con conexión wifi.
- Se realizó una investigación detallada de cada elemento electrónico para realizar el análisis de calidad del agua, por lo que se colocaron tres sensores para realizar las mediciones de los cuatro parámetros de calidad del agua (pH, temperatura, conductividad eléctrica y oxígeno disuelto). Debido a que cada sensor posee limitaciones, se realizaron pruebas con agua de uso residencial y también con agua mezclada con un poco de azúcar (este último para verificar el uso del medidor de conductividad eléctrica).
- Se desarrolló un sistema de adquisición de datos en Python para la visualización de los parámetros realizados por el vehículo mediante la unión de las plataformas Raspberry Pi y Arduino.

RECOMENDACIONES

- El vehículo de superficie no tripulado tiene la capacidad de navegar a un tiempo no mayor a treinta minutos debido a que la tarjeta raspberry consume mucha energía. Por lo que se recomienda que el vehículo deba ser utilizado bajo las siguientes condiciones: primero, debe tener acceso a conexión wifi porque utiliza una dirección IP del router y segundo, las baterías deberán estar cargadas al 100 % para que el vehículo soporte la navegación durante el período de media hora.
- Se debe tener en cuenta que el sensor GPS responde a los 30 segundos de ser encendido el prototipo y si es que no hay un buen clima, el sensor jamás indicará la velocidad del vehículo.
- Para apreciar las distancias en un archivo CAD, el recorrido del vehículo debe ser extenso superior a los 100 metros. Caso contrario, se debe seleccionar los datos de la tabla ubicada en la aplicación Python y colocarlos en un archivo xls para determinar la distancia recorrida por el vehículo.
- El medidor de oxígeno disuelto tiene un tiempo de demora para determinar la lectura real de ppm (partición por millón). Por lo tanto, para realizar la lectura de los tres sensores se recomienda colocar al menos dos minutos y luego guardar los datos en la pantalla de visualización.
- Una desventaja en la medición de sensores es que se debe distribuir la extracción del agua en tres proporciones, por lo que se recomienda en un futuro construir o implementar un sensor multiparámetro que contenga las cuatro señales (pH, temperatura, oxígeno disuelto y conductividad eléctrica).
- Como se ha mencionado anteriormente, el vehículo opera con wifi, esto puede cambiar para trabajos futuros ya que no es el único método para un control inalámbrico, por ejemplo, se puede recurrir a la radiofrecuencia.
- El presente prototipo ayudará a dar una idea para la toma de muestra a lagos, en el sentido de que cuando se requiera ese tipo de muestras, el barco deberá incrementar el tamaño de sus componentes mecánicos y electrónicos.

BIBLIOGRAFÍA

- 330ohms. (09 de febrero de 2016). *Motores a pasos... ¿unipolares o bipolares?* Recuperado el 14 de octubre de 2018, de <https://blog.330ohms.com/2016/02/09/motores-a-pasos-unipolares-o-bipolares/>
- AliExpress. (2018). *Juegos de control remoto*. Recuperado el 20 de noviembre de 2018, de <https://es.aliexpress.com/item/Amphibious-Remote-Control-Car-Hovercraft-RC-Car-Can-Go-Snow-Desert-Land-Surface/32629972826.html>
- Alldatasheet. (2018). *Electronic Components Datasheet*. Recuperado el 25 de noviembre de 2018, de <https://www.alldatasheet.com/>
- Antech. (2011). *Medición de nivel Antech*. Recuperado el 24 de febrero de 2019, de <http://mediciondenivelesantech.blogspot.com/2008/12/medidores-de-nivel-de-liquidos.html>
- Aprendiendoarduino. (2018). *Aprendiendo a manejar Arduino en profundidad*. Recuperado el 15 de octubre de 2018, de <https://www.aprendiendoarduino.com/>
- Aragon, J. O. (2014). *DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA DE GESTIÓN DE UNA RED DE SENSORES APLICADA A LA MONITORIZACIÓN DE LA CALIDAD AMBIENTAL DEL RÍO NAPO*. Lima: Pontificia Universidad Católica del Perú.
- Arduino. (2018). *What is arduino?* Recuperado el 15 de octubre de 2018, de <https://www.arduino.cc/>
- Areatecnologia. (2018). *Motor eléctrico*. Recuperado el 11 de octubre de 2018, de <http://www.areatecnologia.com/>
- Balbuena, J. G. (2017). *DISEÑO DE UN USV PARA EL MONITOREO DE CALIDAD DE AGUA Y TOMA DE MUESTRAS EN LAGOS*. Lima: Pontificia Universidad Católica del Perú.
- Beltrán, D., Palomino, R., Moreno, E., Peralta, C., & Montesinos, D. (2015). *Calidad de agua de la bahía interior de Puno, lago Tititcaca durante el verano del 2011*. Lima: UNMSM.

- Buelta, A., & Martínez, R. (2015). *Guía Básica de Control de Calidad de Agua*. Madrid: ONGAWA.
- Camargo, J. (2009). *Modela de cobertura para redes inalámbricas de interiores*. Sevilla: Escuela Técnica Superior de Ingeniería.
- Carrera, C., & Karol, F. (2001). *Manual de monitoreo: los macroinvertebrados acuáticos como indicadores de la calidad del agua*. Quito: EcoCiencia.
- Chapman, S. (2012). *Máquinas Eléctricas*. Bogotá: Mc Graw Hill.
- CLR. (2016). *Compañía levantina de reductores*. Recuperado el 14 de octubre de 2018, de <https://clr.es/>
- Córdova, J. (2016). Evaluación Integral de la Calidad del Agua de los Embalses y Ríos que conforman el Sistema Hidráulico Chili - Arequipa. Arequipa, Perú.
- Creus, A. (2005). *Instrumentación Industrial*. Barcelona: MARCOMBO S.A.
- Decamaras. (2014). *Es mejor una reflex con sensor CCD o CMOS?* Recuperado el 28 de octubre de 2018, de <https://www.decamaras.com/CMS/content/view/347/40-Es-mejor-una-reflex-con-sensor-CCD-o-CMOS>
- Disetec. (2018). *Soluciones Industriales - Productos y Equipos*. Recuperado el 11 de octubre de 2018, de <http://www.disetec-ec.com/equipos-insumos-soluciones-industriales-instrumentos-medicion-quito-ecuador.php?id=79&t=Instrumentos-Temperatura>
- Diymore. (2018). *PH4502C Value Detection Detect Sensor Module Monitoring Control for Arduino*. Recuperado el 20 de noviembre de 2018, de <https://www.diymore.cc/products/diymore-liquid-ph-value-detection-detect-sensor-module-monitoring-control-for-arduino-m>
- El-Khoury, N. (2016). *ADAPTACIÓN E IMPLEMENTACIÓN DE UN SISTEMA AUTÓNOMO DE BAJO COSTE DE MONITORIZACIÓN DE CALIDAD DEL AGUA EN TIEMPO REAL*. Madrid: Universidad Politécnica de Madrid.
- Eltrinauta. (2016). *Accesorios náuticos, Electrónica náutica, Embarcaciones neumáticas, ropa y calzado*. Recuperado el 15 de octubre de 2018, de <http://www.eltrinauta.com/>

- Embention. (2018). *USV (Unmanned surface vehicle), Aplicaciones y Ventajas*. Recuperado el 10 de octubre de 2018, de <https://www.embention.com/es/news/usv-sus-aplicaciones-y-ventajas/>
- Endress+Hauser. (2018). *Análisis de líquidos*. Recuperado el 11 de octubre de 2018, de <https://www.es.endress.com/es/instrumentacion-campo/analisis-agua-liquidos-industria>
- Ercilio, F., Rodríguez, S., Cabel, W., Ortiz, I., Noriega, P., & Tejada, M. (2005). *Desafíos del derecho humano al agua en el Perú*. Lima: Gráfica Loro's S.A.
- Espartza, I. (2017). *Diseño de un algoritmo de path planning para un drone acuático de superficie basado en grafos Eulerianos*. Sevilla: Universidad de Sevilla.
- Espinosa, R. (03 de septiembre de 2016). *Control de motores CC y CA, servomotores y motores de paso*. Recuperado el 11 de octubre de 2018, de https://www.scribd.com/document_downloads/direct/331844505?extension=pdf&ft=1539295914<=1539299524&show_pdf=true&user_id=183964806&uahk=oFT0zpTQP1G7NILWdNYCZoUV1WY
- Eureka. (2017). *Multiprobes*. Recuperado el 18 de septiembre de 2018, de <https://www.waterprobes.com/multiprobes-and-sondes-for-monitori>
- FAO. (2016). *Organización de las Naciones Unidas para la Alimentación y la Agricultura*. Recuperado el 15 de septiembre de 2018, de http://www.fao.org/nr/water/aquastat/water_use/indexesp.stm
- Fernández, A. (30 de marzo de 2011). *Proyecto de Modernización de la Gestión de los Recursos Hídricos*. Recuperado el 18 de septiembre de 2018, de <http://www.ana.gob.pe/media/361356/3%20protocolo%20nacional%20de%20monitoreo%20af.pdf>
- Forero, N. (2012). Normas de comunicación en serie RS-232, RS-422 y RS-485. *Ingenio Libre, Art13*(Revisa 11), 9.
- Gandia, R. (2004). *Servomotores*. Recuperado el 14 de octubre de 2018, de <https://dscs.webcindario.com/>

- González, C. (2015). *Análisis técnico de los diferentes tipos de baterías comercialmente disponibles para su integración en el proyecto de una microred aislada*. Bogotá: Universidad Distrital Francisco José de Caldas.
- Guerrero, B. (2011). *Equilibrio de los cuerpos flotantes*. Valparaíso: Humboldt Marine Training.
- Hurtado, J. (2012). *Introducción a las redes de comunicación industrial*. Recuperado el 18 de noviembre de 2018, de http://www.infopl.net/files/documentacion/comunicaciones/infoPLC_net_introduccion3b3n-a-las-redes-de-comunicacion3b3n-industrial.pdf
- Ianver. (13 de diciembre de 2010). *Scribd - Contaminación del agua en el Perú*. Recuperado el 17 de septiembre de 2018, de <https://es.scribd.com/doc/45205788/Contaminacion-Del-Agua-en-El-Peru>
- Ies, A. (2018). *Tecnología de la comunicación*. Obtenido de <http://www.ieslaaldea.com/documentos/tecnologia/tecnocomunic01.pdf>
- Innova. (2010). *Oceanografía Litoral*. Recuperado el 20 de noviembre de 2018, de <http://www.oceanografialitoral.com/productos/auv-rov>
- Instructables. (2018). *Comparison of temperature sensors*. Recuperado el 11 de octubre de 2018, de <https://www.instructables.com/id/Comparison-of-Temperature-Sensors/>
- Jumapam. (2018). *Junta Municipal de Agua Potable y Alcantarillado de Mazatlán*. Recuperado el 15 de septiembre de 2018, de <http://jumapam.gob.mx/>
- Larsson, L., & Eliasson, R. (2000). *Principles of Yach Design*. Londres: Adlard Coles Nautical.
- Linares, V., Tapia, A., & Chávez, G. (marzo de 2017). *La Agenda para el Desarrollo de Arequipa 2015 - 2018*. Recuperado el 17 de septiembre de 2018, de <http://ucsp.edu.pe/wp-content/uploads/2017/03/Agenda-para-el-Desarrollo-de-Arequipa-2015-2018-final.pdf>
- López, J. (febrero de 2014). *Fundamentos de los sistemas de radionavegación*. Recuperado el 15 de octubre de 2018, de

http://openaccess.uoc.edu/webapps/o2/bitstream/10609/77345/6/Sistemas%20de%20radionavegaci%C3%B3n_M%C3%B3dulo%203_Fundamentos%20de%20los%20sistemas%20de%20radionavegaci%C3%B3n.pdf

Maraña, J. C. (2005). *Instrumentación y control de procesos*. Bilbao: Idom.

Marine, A. (2018). *Monitor de oxígeno PINPOINT® II*. Recuperado el 20 de noviembre de 2018, de <https://americanmarineusa.com/products/pinpoint-ii-oxygen-monitor>

MINAM. (19 de Diciembre de 2015). *Estándares Nacionales de Calidad Ambiental para Agua*. Obtenido de <http://sinia.minam.gob.pe/normas/estandares-nacionales-calidad-ambiental-agua-0>

Minaya, R. J. (2017). *PARÁMETROS FÍSICOS, QUÍMICOS, MICROBIOLÓGICOS PARA DETERMINAR LA CALIDAD DEL AGUA EN LA LAGUNA MORONOCOCHA, ÉPOCA DE TRANSICIÓN CRECIENTE-VACIANTE. IQUITOS. PERÚ. 2016*. Iquitos: Universidad Nacional de la Amazonía Peruana.

Mokey, I. (2018). *Water quality & instrumentation*. Recuperado el 20 de noviembre de 2018, de <https://www.mokeyinternational.com/collection/?cid=4989323011>

Navegar. (10 de diciembre de 2014). *BeiDou, el nuevo sistema de posicionamiento marítimo de China*. Recuperado el 15 de octubre de 2018, de <http://www.navegar.com/beidou-el-nuevo-sistema-de-posicionamiento-maritimo-de-china/>

Naylamp. (2018). *Naylamp mechatronics*. Obtenido de <https://naylampmechatronics.com/>

Ocola, J. (24 de marzo de 2010). *Protección del agua - vigilancia y control de vertimientos - Paver*. Recuperado el 17 de septiembre de 2018, de <http://www.ana.gob.pe/media/353227/4-protecci%C3%B3n%20del%20agua%20vigilancia%20y%20control%20de%20vertimientos%20paver.%20%20lic.%20juan%20ocola.pdf>

OEFA. (febrero de 2015). *Instrumentos Básicos para la Fiscalización Ambiental*. Recuperado el 17 de septiembre de 2018, de https://www.oefa.gob.pe/?wpfb_dl=13978

- Omega. (2018). *Omega a sprectis company*. Recuperado el 18 de septiembre de 2018, de <https://cl.omega.com>
- Parra, L. (2012). *Microprocesadores*. Tlalnepantla de Baz: Red Tercer Milenio S.C.
- PCE. (30 de septiembre de 2014). *Equipos de medida - Balanzas - Regulación y control*. Recuperado el 16 de septiembre de 2018, de <https://www.pce-iberica.es/>
- Proaño, G. (2009). *Estudio técnico comparativo de redes lan alámbricas e inalámbricas*. Quito: Pontificia Universidad Católica del Ecuador.
- Quecamaraflex. (2018). *Cómo funciona el sensor de una cámara digital*. Recuperado el 15 de octubre de 2018, de <http://quecamarareflex.com/como-funciona-el-sensor-de-una-camara-digital/>
- Quijarro, L. (2015). *Empleo de Vehículos no Tripulados en la investigación Científica Antártica*. Recuperado el 17 de septiembre de 2018, de <https://revistamarina.cl/revistas/2015/1/lquijarros.pdf>
- Ramírez, J. L. (2018). *DISEÑO DE PLATAFORMA DE ACOPLE Y MINIROV PARA VEHÍCULOS MARINOS DE SUPERFICIE*. Lima: Pontificia Universidad Católica del Perú.
- Raspberrypi. (2018). *Frambuesa pi*. Recuperado el 15 de octubre de 2018, de <https://www.raspberrypi.org/>
- Rebassa, M., & Castells, M.-l. (2016). *Estudio y Diseño de una Embarcación de Recreo*. Barcelona: Universidad Politécnica de Cataluña.
- Rivera, R. E. (2012). *MEDICIÓN Y CONTROL TELEMÉTRICO DE APROVECHAMIENTOS DE AGUAS NACIONALES Y DESCARGAS RESIDUALES*. México: Universidad Nacional Autónoma de México.
- Sánchez, C. (2016). *Fragmentación de Internet*. Recuperado el 10 de 20 de 2018, de https://intranet.bibliotecasgc.bage.es/intranet-tmpl/prog/local_repository/documents/18011.pdf
- Sánchez, D. (2016). *Calidad de las aguas*. Toledo: Universidad de Castilla - La Mancha.

- Sepúlveda, R. T. (2011). *DISEÑO DE MODELOS DE CALIDAD DEL AGUA MEDIANTE EL USO DE LA PERCEPCIÓN REMOTA*. Embalse Valle de Bravo: Universidad Nacional Autónoma de México.
- SolerPalau. (2015). *Mecánica de fluidos: Velocidad del aire*. Obtenido de <http://www.solerpalau.mx/pdf/hojas-tecnicas/velocidad-aire.pdf>
- Tejada, E. G. (2017). *VEHÍCULO AUTÓNOMO DE SUPERFICIE PARA REALIZAR ESTUDIOS BATIMÉTRICOS*. Lima: Pontificia Universidad Católica del Perú.
- TuTiempo. (2019). *Calcular distancias*. Obtenido de <https://www.tutiempo.net/calcular-distancias.html>
- UTFSM. (12 de julio de 2010). *Universidad Técnica Federico Santa María - Departamento de Electrónica*. Recuperado el 15 de octubre de 2018, de <http://www2.elo.ut fsm.cl/~elo383/apuntes/PresentacionBaterias.pdf>
- Vimael. (2018). *Empresa importadora especializada en equipos de medición y seguridad*. Recuperado el 20 de noviembre de 2018, de <http://www.vimael.com/index.php/medidores-sobre-agua-tierra-ph-tds-orp-conductividad-y-otros/569-medidor-de-conductividad-marca-va-modelo-1022-con-sensor-interno.html>
- Zamora, A., Rustrian, E., Sánchez, B., & González, G. (2017). *TECNOLOGÍA SIG PARA EL MONITOREO DE LA CALIDAD DEL AGUA EN VERACRUZ*. Veracruz: Universidad Veracruzana.
- Zima, R. (2018). *Robótica Móvil*. Recuperado el 10 de octubre de 2018, de <https://www.zimarobotics.com/areas-de-negocio/vehiculos-no-tripulados-drones/>

ANEXOS

ANEXO 01. MANUAL DE USUARIO – VEHÍCULO DE SUPERFICIE NO TRIPULADO

ANEXO 02. DISPOSITIVOS DEL VEHÍCULO DE SUPERFICIE NO TRIPULADO

ANEXO 03. CÓDIGO DE LA TARJETA ARDUINO NANO

ANEXO 04. CÓDIGO DE LA TARJETA RASPBERRY PI3 B+

ANEXO 05. MEDIDOR DE OXÍGENO PINPOINT II

ANEXO 06. MEDIDOR DE CONDUCTIVIDAD ELÉCTRICA VA 1022

ANEXO 07. SENSOR DE PH Y TEMPERATURA

ANEXO 08. SENSOR DE FLUJO

ANEXO 09. PASOS DE INSTALACIÓN DE LA PLATAFORMA ARDUINO

ANEXO 10. PASOS DE INSTALACIÓN DE LA PLATAFORMA RASPBERRY

ANEXO 11. COSTO DEL VEHÍCULO DE SUPERFICIE NO TRIPULADO PARA MEDIR LA CALIDAD DEL AGUA

ANEXO 12. PLANOS ELECTRÓNICOS A LA PLATAFORMA ARDUINO NANO

ANEXO 13. PLANOS ELECTRÓNICOS A LA PLATAFORMA RASPBERRY PI3 B+

ANEXO 14. PLANOS MECÁNICOS - ENSAMBLE DEL USV

ANEXO 15. PLANOS MECÁNICOS - TAPA SUPERIOR

ANEXO 16. PLANOS MECÁNICOS - BRAZO

ANEXO 17. PLANOS MECÁNICOS - ROSCA

ANEXO 18. PLANOS MECÁNICOS - TAPA DELANTERA

ANEXO 19. PLANOS MECÁNICOS - TAPA TRASERA

ANEXO 20. PLANOS MECÁNICOS - TAPÓN

ANEXO 21. PLANOS MECÁNICOS - ACOPLE

ANEXO 22. PLANOS MECÁNICOS - CODO

ANEXO 01. MANUAL DE USUARIO – VEHÍCULO DE SUPERFICIE NO TRIPULADO

El prototipo consta de tres partes: vehículo de superficie no tripulado (Figura A.1.1), una caja la cual es utilizada para realizar las mediciones de los parámetros de calidad del agua (Figura A.1.2) y la visualización de los parámetros utilizando un computador con la aplicación VNC Viewer, visto en la Figura A.1.3



Figura A.1.1 Vehículo de superficie no tripulado

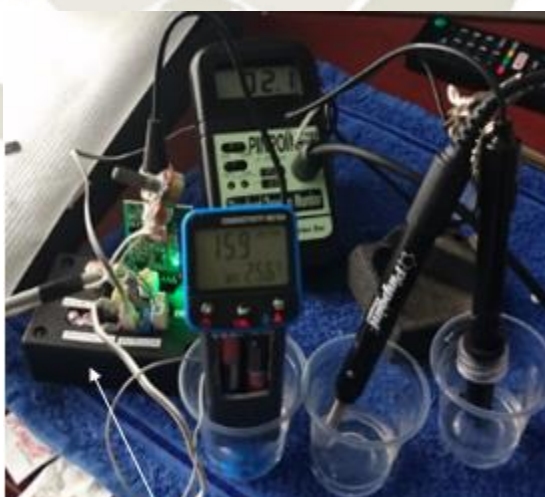


Figura A.1.2 Caja de sensores para calidad del agua

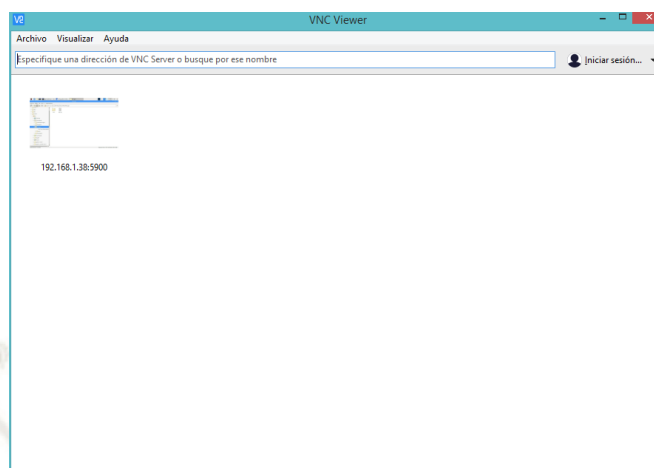


Figura A.1.3 Vnc Viewer

Para un correcto funcionamiento, se debe seguir los siguientes pasos:

1. Verificar que estén cargadas las dos fuentes de alimentación.
2. Verificar que estén conectados los cables de la Power Bank a la Raspberry Pi y del Arduino Nano a la Raspberry Pi, visto en la Figura A.1.4.

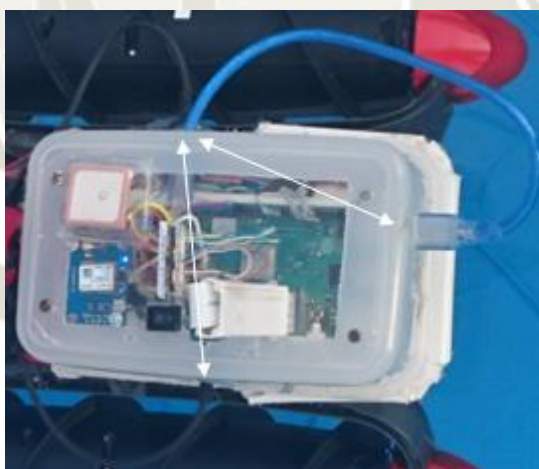


Figura A.1.4 Cableado

3. Verificar que el interruptor ubicado encima del vehículo esté apagado, visto en la Figura A.1.5.



Figura A.1.5 Encerado de sensores al Arduino

4. Dar clic en el acceso directo (app.py) y ejecutarlo, visto en la Figura A.1.6.

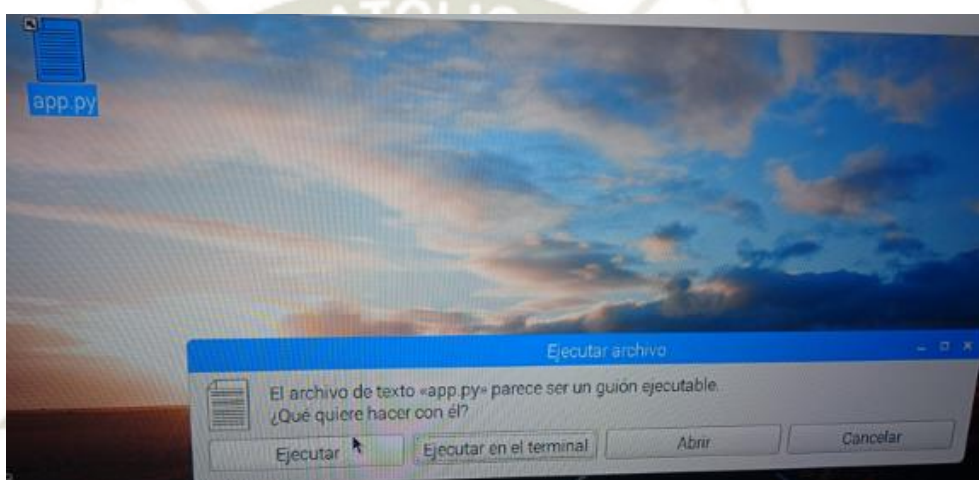


Figura A.1.6 Inicio de programa

5. Una vez abierto el programa (Figura A.1.7), encender el interruptor para la activación de los motores, visto en la Figura A.1.8.

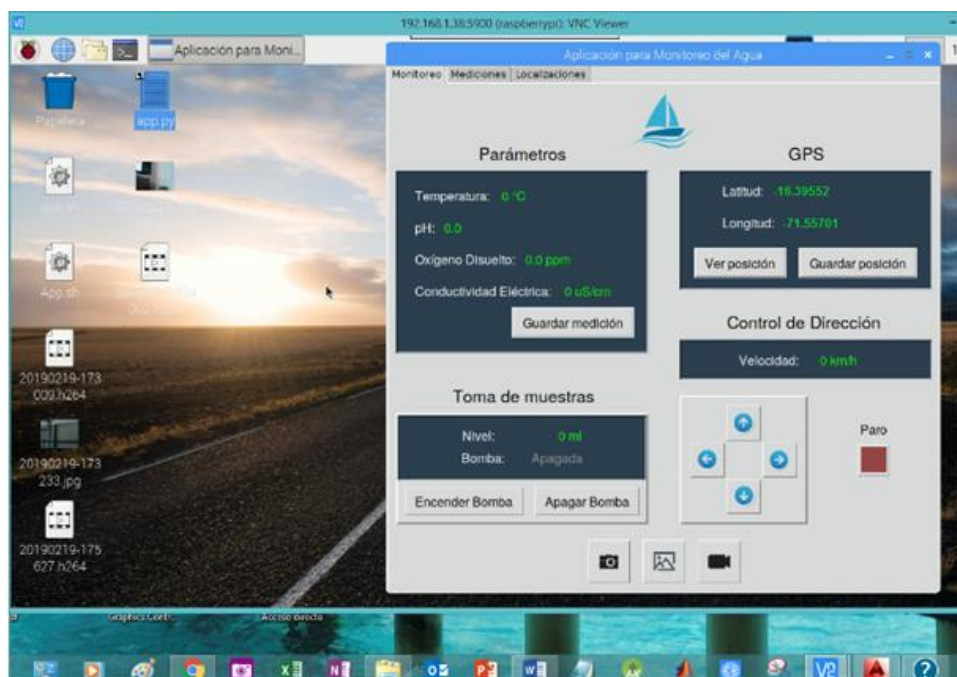


Figura A.1.7 Programa Python



Figura A.1.8 Activación de la fuente para los motores

6. En caso de no conectar correctamente el cable usb de la Raspberry Pi al Arduino Nano, se muestra un mensaje tal como indica la Figura A.1.9.

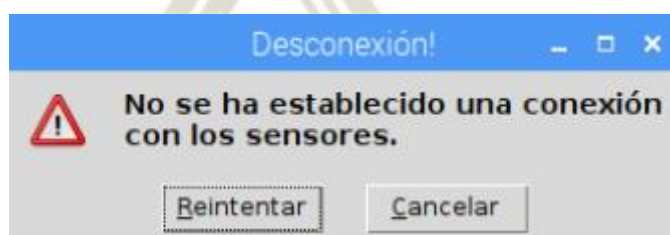


Figura A.1.9 Desconexión entre Raspberry Pi y Arduino Nano

7. Se procede a mover el vehículo de superficie no tripulado y cada vez se puede ir guardando las posiciones que indica el GPS (Figura A.1.7), se llega a un punto

deseado y se extrae el líquido, en caso de no ver la cantidad a llenar automáticamente se apaga la bomba cuando en la Raspberry Pi capta los 25 mililitros (Figura A.1.10)

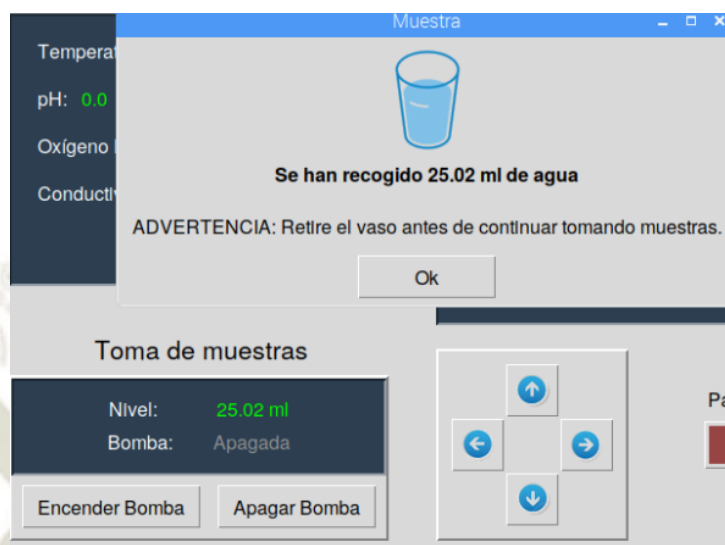


Figura A.1.10 Extracción del líquido

8. Se regresa al punto de origen, sacamos la muestra del vehículo y se coloca otro vaso para que el vehículo repita el paso anterior. La caja de sensores debe estar conectado al vehículo. Al final cada muestra de líquido se debe repartir en tres vasos para que cada sensor realice las respectivas mediciones (Figura A.1.11).




Figura A.1.11 Conexión y medición de los sensores

9. Por último, se visualizan las mediciones de los parámetros de calidad del agua (Figura A.1.12) y las posiciones del GPS en una pequeña base de datos (Figura A.1.13)

Aplicación para Monitoreo del Agua

Monitoreo Mediciones Localizaciones



Mediciones realizadas


No.	Temperatura	Oxígeno Disuelto	pH	Conductividad	Fecha y Hora
1	19.0	0.0	7.0	113.0	2019-02-27 16:48:02.1
2	20.0	1.2	7.2	123.0	2019-02-27 16:48:55.8
3	20.0	2.6	7.2	123.0	2019-02-27 16:51:08.0
4	19.0	1.6	7.2	132.0	2019-02-27 16:52:48.4

Borrar mediciones

Figura A.1.12 Parámetros de calidad del agua

Aplicación para Monitoreo del Agua

Monitoreo Mediciones Localizaciones



Localizaciones almacenadas

No.	Latitud	Longitud	Fecha y Hora
1	-16.39552	-71.55701	2019-03-01 15:57:11.60637
2	-16.39551	-71.55702	2019-03-01 15:57:21.41071
3	-16.39551	-71.557	2019-03-01 15:57:29.59859
4	-16.39552	-71.55695	2019-03-01 15:57:56.78263
5	-16.39552	-71.55694	2019-03-01 15:58:03.74816
6	-16.39553	-71.55694	2019-03-01 15:58:11.37124
7	-16.39552	-71.55693	2019-03-01 15:58:14.20770
8	-16.39551	-71.55694	2019-03-01 15:58:17.72973
9	-16.39549	-71.55696	2019-03-01 15:58:23.60744
10	-16.3955	-71.55696	2019-03-01 15:58:29.60251

Borrar posiciones

Figura A.1.13 Posiciones del GPS

En caso de requerir la cámara, fotografías o videos en tiempo real se pueden activar y desactivar los botones vistos en la Figura A.1.14.

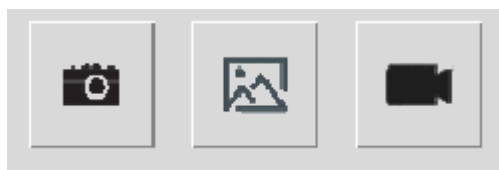


Figura A.1.14 Cámara, fotografía o video en tiempo real

Para la carga de baterías se debe abrir la caja plástica y colocar los cargadores.



ANEXO 02. DISPOSITIVOS DEL VEHÍCULO DE SUPERFICIE NO TRIPULADO

En las siguientes Figuras se visualizan como está formado el vehículo de superficie no tripulado.



Figura A.2.1 Batería para los motores



Figura A.2.2 Elementos del vehículo de superficie no tripulado



Figura A.2.3 Conexiones parte 1

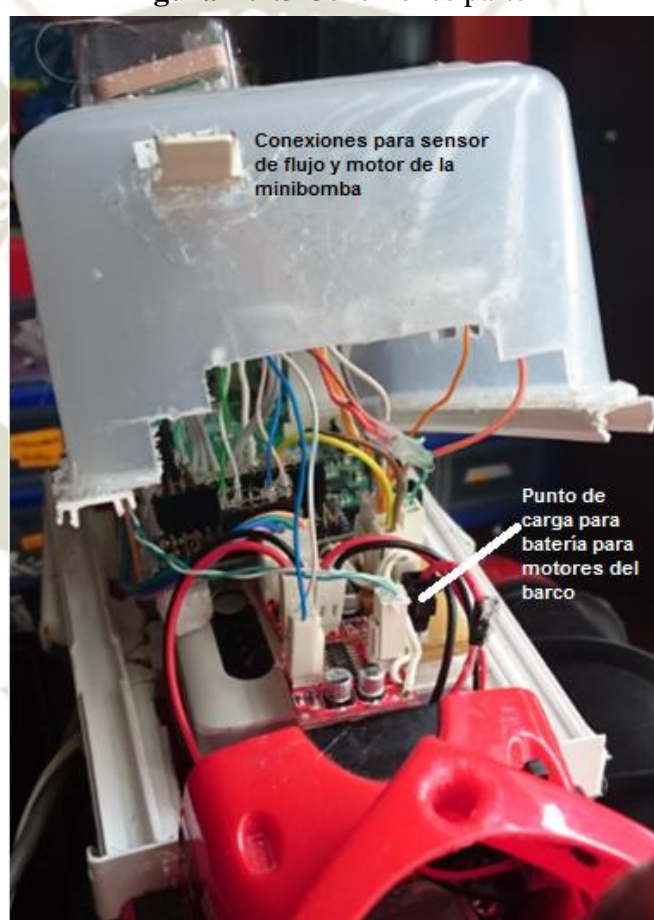


Figura A.2.4 Conexiones parte 2

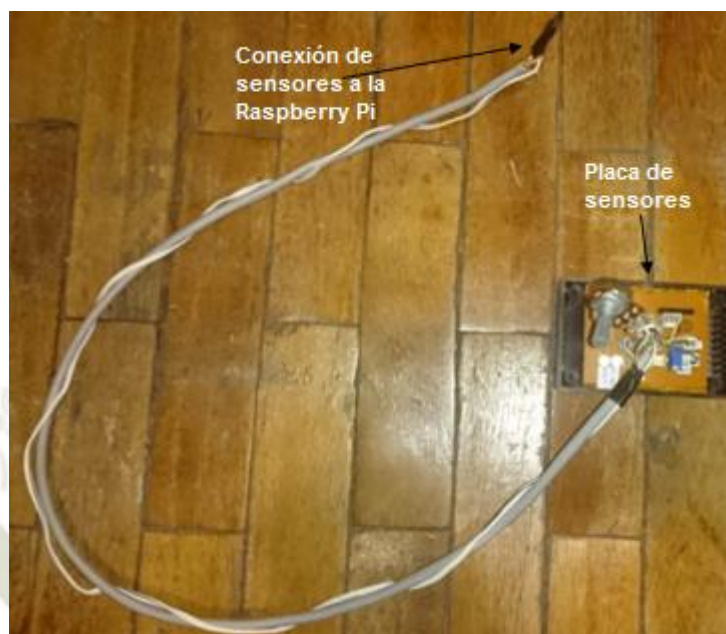


Figura A.2.5 Conexiones parte 3

ANEXO 03. CÓDIGO DE LA TARJETA ARDUINO NANO

```
#include <String.h>
//Declarar variables de los sensores para calidad del agua
float ph=0;
float temperatura=0;
float OD1=0;
float OD;
float CE=0;
int CEPot=0;
int encender=0;
//Declarar variables para el sensor de Flujo
volatile int Pulsos; //variable para la cantidad de pulsos recibidos
int Fin = 2; //Sensor conectado en el pin 2
float factor=7.5; //para convertir de frecuencia a caudal
float caudal_mLseg=0;
//Declarar variables selección del sensor CE
const int timeThreshold = 50;
const int intPin = 3;
volatile int ISRCCounter = 0;
int counter = 0;
long timeCounter=0;
const int led1=7;
const int led2=6;
int cond;

void setup() {
  pinMode(intPin, INPUT); //Selector CE
  pinMode(Fin, INPUT); //Flujo
  pinMode(led1, OUTPUT); //Indicador 1
  pinMode(led2, OUTPUT); //Indicador 2
  Serial.begin(9600); //Iniciamos el puerto serie
  attachInterrupt(0, ContarPulsos, RISING); // (Interrupcion 0 (Pin2), funcion, Flanco de
  subida)
  attachInterrupt(digitalPinToInterrupt(intPin), debounceCount, LOW);
}
void ContarPulsos()
{
  Pulsos++;
}
//---Función para obtener frecuencia de los pulsos-----
int ObtenerFrecuencia()
{
  int frecuencia;
  Pulsos = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
```



```

delay(10); //muestra de 1 segundo
noInterrupts(); //Desabilitamos las interrupciones
frecuencia=Pulsos; //Hz(pulsos por segundo)
return frecuencia;
}
void debounceCount()
{
    if (counter == 4){
        ISRCounter = 0;
    }
    if(millis()>timeCounter + timeThreshold){
        ISRCounter++;
        timeCounter=millis();
    }
}
void loop()
{
    float frecuencia=ObtenerFrecuencia(); //obtenemos la Frecuencia de los pulsos en Hz
    //caudal_mLseg=frecuencia*16.67/(5.4*factor); //calculamos el caudal en mL/seg
    if (frecuencia > 0){
        caudal_mLseg = 0.0816;
    }
    encender = analogRead (A5);
    if (encender < 100){ //Cuando está apagado el interruptor
        temperatura = 0;
        ph = 0;
        CE = 0;
        OD = 0;
    }
    else {
        OD=0;
        ph=0;
        CE=0;
        temperatura = analogRead (A0)/37.7;
        delay(100);
        for (int i=0; i<20; i++){
            ph += analogRead (A1)/121.42;
            delay(50);
        }
        ph/=20;
        for (int i=0; i<20; i++){
            OD += (0.121*analogRead (A2))-56.3;
            delay(100);
        }
        OD/=20;
        if (OD < 0.1){
            OD =0;

```

```

}
if (counter != ISRCounter){
    counter = ISRCounter;
    if (counter == 1){
        digitalWrite(led1,LOW);
        digitalWrite(led2,HIGH);
    }
    else if (counter == 2){
        digitalWrite(led1,HIGH);
        digitalWrite(led2,LOW);
    }
    else if (counter == 3){
        digitalWrite(led1,HIGH);
        digitalWrite(led2,HIGH);
    }
    else{
        digitalWrite(led1,LOW);
        digitalWrite(led2,LOW);
    }
}
if (counter == 1){
    for (int i=0; i<20; i++){
        CEPot += analogRead (A4);
        delay(100);
    }
    CEPot/=20;
    CE=map(CEpot, 0, 1023, 0, 2000); //convierte el valor del potenciómetro a uS/cm
}
else if (counter == 2){
    for (int i=0; i<20; i++){
        CE += (0.8908*analogRead (A3))-105.47; //Ecuación del Agua de la Llave
        delay(100);
    }
    CE/=20;
}
else if (counter == 3){
    for (int i=0; i<20; i++){
        CE += (0.8395*analogRead (A3))+53.754; //Ecuación del Agua + Azúcar
        delay(100);
    }
    CE/=20;
}
else{
    CE=0;
}
}
Serial.print (" ");

```

```
Serial.print (temperatura,0);  
Serial.print (" : ");  
Serial.print (ph,1);  
Serial.print (" : ");  
Serial.print (OD,1);  
Serial.print (" : ");  
Serial.print (CE,0);  
Serial.print (" : ");  
Serial.print (caudal_mLseg,4);  
Serial.println (" : ");  
}
```



ANEXO 04. CÓDIGO DE LA TARJETA RASPBERRY PI3 B+

```
//PANTALLA PRINCIPAL
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import os
import tkinter as tk
from frames.main import MainFrame
from frames.data import DataFrame
from frames.menu import MenuNotebook
from models.database import Database
from styles.main_styles import Style
class Application(tk.Frame):

    """ APLICACIÓN PARA MONITOREO DE AGUA """
    def __init__(self, parent, *args, **kwargs):
        tk.Frame.__init__(self, parent, *args, **kwargs)
        self.parent = parent
        self.init_gpsd()
        self.init_window()
        self.parent.protocol("WM_DELETE_WINDOW", self.destroy_gpsd)
        self.pack(side=tk.TOP, fill=tk.BOTH, expand=1)
        # Notebook pages (tabs)
        self.notebook = MenuNotebook(self)
        main_tab = MainFrame(self.notebook, width=self.window_width,
                              height=self.window_height)
        mediciones_tab = DataFrame(self.notebook,
                                   name='mediciones',
                                   width=self.window_width,
                                   height=self.window_height)
        posiciones_tab = DataFrame(self.notebook,
                                   name='posiciones',
                                   width=self.window_width,
                                   height=self.window_height)
        self.notebook.add(main_tab, text="Monitoreo")
        self.notebook.add(mediciones_tab, text="Mediciones")
        self.notebook.add(posiciones_tab, text="Localizaciones")
        self.notebook.place(x=0, y=0)

    def init_window(self):
        """ MÉTODO PARA CONFIGURAR LA VENTANA """
        self.parent.title("Aplicación para Monitoreo del Agua")
        self.window_width = 700
        self.window_height = 640
        self.parent.geometry('{}x{}'.format(self.window_width,
                                              self.window_height+22))
        self.parent.resizable(width=False, height=False)
        self.parent.config(background=Style.BACKGROUND_COLOR)
```

```
def init_gpsd(self):
    """INICIALIZACIÓN DEL DAEMON GPSD"""
    os.system('sudo killall gpsd')
    os.system('sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock')

def destroy_gpsd(self):
    """Paro del daemon gpsd"""
    os.system('sudo killall gpsd')
    self.parent.quit()

if __name__ == '__main__':
    root = tk.Tk()
    Database.create_tables()
    app = Application(root)
    app.mainloop()

-----

//PROGRAMA PRINCIPAL
# -*- coding: utf-8 -*-
import itertools as it
import time
from decimal import Decimal
import tkinter as tk
import webbrowser
from tkinter import messagebox, ttk
from gps3.agps3threaded import AGPS3mechanism

import serial
from serial.serialutil import SerialException

from styles.main_styles import Style

from classes.drone import Drone
from dialogs.sample_dialog import SampleDialog

from picamera import PiCamera
camera = PiCamera()

class MainFrame(tk.Frame):
    """
    PESTAÑA PRINCIPAL QUE CONTIENE 4 SECCIONES, SENSORES (DONDE
    SE MIDE TODOS LOS PARÁMETROS DEL AGUA), GPS (EN DONDE SE
    OBTIENE INFORMACIÓN DE LA LATITUD Y LONGITUD), CONTROLES
```

(PARA EL CONTROL DIRECCIONAL DEL DRON ACUÁTICO) Y BOMBA
(DONDE SE CONTROLA LA BOMBA DE SUCCIÓN DE AGUA)

"""

```
def __init__(self, parent, *args, **kwargs):
    tk.Frame.__init__(self, parent, *args, **kwargs)
    self.parent = parent

    self.serial_conn = False
    self.drone = Drone()

    # agps thread
    self.agps_thread = AGPS3mechanism()
    self.agps_thread.stream_data()
    self.agps_thread.run_thread()

    # Valores de los sensores
    self.temperatura = tk.StringVar()
    self.ph = tk.StringVar()
    self.oxigeno = tk.StringVar()
    self.conductividad = tk.StringVar()

    self.latitud = tk.StringVar()
    self.longitud = tk.StringVar()

    self.velocidad = tk.StringVar()
    self.nivel = tk.StringVar()
    self.bomba = tk.StringVar()

    self.nivel.set('0 ml')
    self.nivel_totalizado = 0.0
    self.time_start_pump = 0.0

    if self.drone.bomba:
        self.bomba.set('Encendida')
    else:
        self.bomba.set('Apagada')

    # Secciones de la aplicación
    self.seccion_sensores()
    self.seccion_gps()
    self.seccion_controles()
    self.seccion_bomba()
    self.seccion_camara()

def init_serial(self):
    try:
```



```

self.progress_dialog = tk.Toplevel()
self.progress_dialog.title('Realizando conexiÃ³n')
self.progress_dialog.transient(self)

info_message = 'Conectando...'
message_label = tk.Label(
    self.progress_dialog, text=info_message,
    font=Style.POPUP_FONT)
message_label.pack(padx=10, pady=5)

progressbar = ttk.Progressbar(
    self.progress_dialog, mode="indeterminate", length=300)
progressbar.pack(padx=5, pady=5)
progressbar.start(8)

# Serial Buffer
self.ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=0,
    writeTimeout=0) # asegura el no bloqueo
self.serial_buffer = ""
except SerialException:
    self.progress_dialog.destroy()
    self.serial_conn = False
    retry = messagebox.askretrycancel(
        'DesconexiÃ³n!', 'No se ha establecido una conexiÃ³n con los '
        'sensores.'
    )
    if retry:
        self.init_serial()
    else:
        self.parent.quit()

else:
    self.serial_conn = True

def seccion_sensores(self):
    """SECCIÓN DONDE SE MUESTRA LOS PARÁMETROS DEL AGUA."""
    # Posicionamiento
    sensores_y_pos = 70
    sensores_x_pos = 10

    # Header sección de sensores
    sensores_header = tk.Label(
        self, text="Parámetros",
        font=Style.HEADER_FONT)
    sensores_header.place(x=sensores_x_pos+102.5, y=sensores_y_pos)

    # Frame para sección de sensores

```

```

sensores_frame = tk.Frame(self,
                            borderwidth=2, relief="sunken",
                            width=320, height=230,
                            bg=Style.PRIMARY_COLOR)
sensores_frame.place(x=sensores_x_pos, y=sensores_y_pos+40)

# Posicionamiento de etiquetas dentro del frame
labels_y_pos = 15
labels_x_pos = 20

temperatura_label = tk.Label(
    sensores_frame, text="Temperatura:",
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)
temperatura_label.place(x=labels_x_pos, y=labels_y_pos)
ph_label = tk.Label(
    sensores_frame, text="pH:",
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)
ph_label.place(x=labels_x_pos, y=labels_y_pos+40)
oxigeno_label = tk.Label(
    sensores_frame, text="Oxígeno Disuelto:",
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)
oxigeno_label.place(x=labels_x_pos, y=labels_y_pos+80)
conductividad_label = tk.Label(
    sensores_frame, text="Conductividad Eléctrica:",
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)
conductividad_label.place(x=labels_x_pos, y=labels_y_pos+120)

# Valores de los sensores
self.temperatura_output = tk.Label(
    sensores_frame, textvariable=self.temperatura,
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
    fg=Style.DISPLAY_COLOR)
self.temperatura_output.place(x=labels_x_pos+110, y=labels_y_pos)
self.ph_output = tk.Label(
    sensores_frame, textvariable=self.ph,
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
    fg=Style.DISPLAY_COLOR)
self.ph_output.place(x=labels_x_pos+37, y=labels_y_pos+40)
self.oxigeno_output = tk.Label(
    sensores_frame, textvariable=self.oxigeno,
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
    fg=Style.DISPLAY_COLOR)
self.oxigeno_output.place(x=labels_x_pos+140, y=labels_y_pos+80)
self.conductividad_output = tk.Label(
    sensores_frame, textvariable=self.conductividad,
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
    fg=Style.DISPLAY_COLOR)

```

```

self.conductividad_output.place(x=labels_x_pos+190, y=labels_y_pos+120)

bomba_guardar_medicion = tk.Button(sensores_frame,
                                    text="Guardar mediciÃ³n",
                                    font=Style.TEXT_FONT,
                                    command=self.check_sensores_data)
bomba_guardar_medicion.place(x=labels_x_pos+125, y=labels_x_pos+150)

self.parent.after(0, self.read_sensors)

def check_sensores_data(self):
    if self.temperatura.get() == "\
    or self.longitud.get() == "\
    or self.ph.get() == "\
    or self.conductividad.get() == ":
        messagebox.showwarning('Sensores no listos!',
                                'La conexi3n a los sensores aÃºn no se ha '
                                'establecido.')
    else:
        self.parent.almacenar_medicion()

def seccion_gps(self):
    """Secci3n donde se muestra la posici3n del dron acuático."""
    # Posicionamiento
    gps_y_pos = 70
    gps_x_pos = 370

    # Header secci3n de gps
    gps_header = tk.Label(
        self, text="GPS",
        font=Style.HEADER_FONT)
    gps_header.place(x=gps_x_pos+135.5, y=gps_y_pos)

    # Frame para secci3n de gps
    gps_frame = tk.Frame(self,
                          borderwidth=2, relief="sunken",
                          width=320, height=150,
                          bg=Style.PRIMARY_COLOR)
    gps_frame.place(x=gps_x_pos, y=gps_y_pos+40)

    # Posicionamiento de etiquetas dentro del frame
    labels_y_pos = 9
    labels_x_pos = 50

    latitud_label = tk.Label(
        gps_frame, text="Latitud:",
        font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)

```



```

latitud_label.place(x=labels_x_pos, y=labels_y_pos)
longitud_label = tk.Label(
    gps_frame, text="Longitud:",
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)
longitud_label.place(x=labels_x_pos, y=labels_y_pos+40)

btn_pos = 0.65

# Botón para guardar posición
self.boton_guardar_posicion = tk.Button(
    gps_frame,
    text="Guardar posición",
    font=Style.TEXT_FONT,
    command=self.check_gps_data
)

self.boton_guardar_posicion.place(relx=0.47, rely=btn_pos)

# Botón para ver posición en navegador
boton_abrir_navegador = tk.Button(gps_frame,
    text="Ver posición",
    font=Style.TEXT_FONT,
    command=self.open_location)
boton_abrir_navegador.place(relx=0.05, rely=btn_pos)

# Valores del sensor
self.latitud_output = tk.Label(
    gps_frame, textvariable=self.latitud,
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
    fg=Style.DISPLAY_COLOR)
self.latitud_output.place(x=labels_x_pos+65, y=labels_y_pos)
self.longitud_output = tk.Label(
    gps_frame, textvariable=self.longitud,
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
    fg=Style.DISPLAY_COLOR)
self.longitud_output.place(x=labels_x_pos+80, y=labels_y_pos+40)

def check_gps_data(self):
    not_allowed = ("", 'n/a')
    if self.latitud.get() not in not_allowed \
        or self.longitud.get() not in not_allowed:
        self.parent.almacenar_posicion()
    else:
        messagebox.showwarning('GPS no listo', 'Los valores del GPS '
            'aún no están listos.')

def seccion_controles(self):

```

```

""SECCIÓN DONDE SE CONTROLA LA DIRECCIÓN DEL DRON""
# Posicionamiento
controles_x_pos = 370
controles_y_pos = 293

# Header sección de controles
controles_header = tk.Label(
    self, text="Control de Dirección",
    font=Style.HEADER_FONT)
controles_header.place(x=controles_x_pos+57.5, y=controles_y_pos-11)

# Frame para sección de controles
controles_frame = ttk.Frame(self,
    borderwidth=2, relief="raised",
    padding=(10, 10),
    width=320, height=320)
controles_frame.place(x=controles_x_pos, y=controles_y_pos+100)

# Frame para mostrar la velocidad del dron
velocidad_frame = tk.Frame(self,
    borderwidth=2, relief="sunken",
    bg=Style.PRIMARY_COLOR)
velocidad_frame.place(
    x=controles_x_pos, y=controles_y_pos+30, width=320, height=50)
# Velocidad del dron
velocidad_label = tk.Label(
    velocidad_frame, text="Velocidad:",
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)
velocidad_label.place(relx=0.35, rely=0.5, anchor=tk.CENTER)

velocidad_output = tk.Label(
    velocidad_frame, textvariable=self.velocidad,
    font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
    fg=Style.DISPLAY_COLOR)
velocidad_output.place(relx=0.63, rely=0.5, anchor=tk.CENTER)

# Botones
up_img = tk.PhotoImage(file="~/Documents/usv/imgs/up.gif")
down_img = tk.PhotoImage(file="~/Documents/usv/imgs/down.gif")
left_img = tk.PhotoImage(file="~/Documents/usv/imgs/left.gif")
right_img = tk.PhotoImage(file="~/Documents/usv/imgs/right.gif")
stop_img = tk.PhotoImage(file="~/Documents/usv/imgs/stop.gif")
self.stop_rec_img = tk.PhotoImage(file="~/Documents/usv/imgs/stop_rec.gif")

button_up = tk.Button(controles_frame, image=up_img,
    command=self.drone.up,
    height=40, width=40)

```

```

button_up.image = up_img
button_up.grid(column=1, row=0)

button_left = tk.Button(controles_frame, image=left_img,
                        command=self.drone.left,
                        height=40, width=40)
button_left.image = left_img
button_left.grid(column=0, row=1)

button_right = tk.Button(controles_frame, image=right_img,
                        command=self.drone.right,
                        height=40, width=40)
button_right.image = right_img
button_right.grid(column=2, row=1)

button_down = tk.Button(controles_frame, image=down_img,
                        command=self.drone.down,
                        height=40, width=40)
button_down.image = down_img
button_down.grid(column=1, row=2)

# stop button and label
stop_x = 0.88
stop_y = 0.65

stop_label = tk.Label(self, text="Paro", font=Style.TEXT_FONT)
stop_label.place(relx=stop_x, rely=stop_y+0.03, anchor=tk.CENTER)

button_stop = tk.Button(self, image=stop_img,
                        command=self.drone.stop)
button_stop.image = stop_img
button_stop.place(x=controles_x_pos+225, y=controles_y_pos+160)

def seccion_bomba(self):
    """Sección donde se controla la bomba de succión de agua"""
    # Posicionamiento
    bomba_x_pos = 10
    bomba_y_pos = 375

    # Header sección de la bomba
    bomba_header = tk.Label(
        self, text="Toma de muestras",
        font=Style.HEADER_FONT)
    bomba_header.place(x=bomba_x_pos+70, y=bomba_y_pos)

    # Frame para sección de bomba
    bomba_frame = tk.Frame(self,

```



```

        borderwidth=2, relief="raised",
        width=320, height=140)
bomba_frame.place(x=bomba_x_pos, y=bomba_y_pos+40)

# Posicionamiento de botones

btns_x_pos = 8
btns_y_pos = 90

# Frame para mostrar el nivel en el vaso
nivel_frame = tk.Frame(bomba_frame,
        borderwidth=2, relief="sunken",
        bg=Style.PRIMARY_COLOR)
nivel_frame.place(relx=0.5, rely=0.01, relwidth=0.999, relheight=0.6,
        anchor=tk.N)
# Nivel del vaso
nivel_label = tk.Label(
        nivel_frame, text="Nivel:",
        font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)
nivel_label.place(relx=0.25, rely=0.3, anchor=tk.W)

nivel_output = tk.Label(
        nivel_frame, textvariable=self.nivel,
        font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
        fg=Style.DISPLAY_COLOR)
nivel_output.place(relx=0.75, rely=0.3, anchor=tk.E)

# Nivel del vaso
bomba_label = tk.Label(
        bomba_frame, text="Bomba:",
        font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR, fg=Style.WHITE)
bomba_label.place(relx=0.25, rely=0.4, anchor=tk.W)

self.bomba_output = tk.Label(
        bomba_frame, textvariable=self.bomba,
        font=Style.TEXT_FONT, bg=Style.PRIMARY_COLOR,
        fg=Style.OFF_COLOR)
self.bomba_output.place(relx=0.75, rely=0.4, anchor=tk.E)

# Botones para control de bomba
bomba_on_btn = tk.Button(bomba_frame, text="Encender Bomba",
        command=self.pump_on,
        font=Style.TEXT_FONT)
bomba_on_btn.place(x=btns_x_pos, y=btns_y_pos)

bomba_off_btn = tk.Button(bomba_frame, text="Apagar Bomba",
        command=self.pump_off,

```

```

font=Style.TEXT_FONT)

bomba_off_btn.place(x=btns_x_pos+165, y=btns_y_pos)

def seccion_camara(self):
    """BOTÓN PARA ACTIVAR O DESACTIVAR LA CÁMARA DE LA
    RASPBERRY PI"""
    camara_img = tk.PhotoImage(file="~/Documents/usv/imgs/camara.gif")
    camara_img_on = tk.PhotoImage(file="~/Documents/usv/imgs/camara_on.gif")
    picture_img = tk.PhotoImage(file="~/Documents/usv/imgs/picture.gif")
    record_img = tk.PhotoImage(file="~/Documents/usv/imgs/record.gif")

    # Positioning
    pos_x = 0.5
    pos_y = 0.94

    self.camaras = it.cycle([camara_img_on, camara_img])
    self.record_imgs = it.cycle([self.stop_rec_img, record_img])
    self.camara_methods = it.cycle([self.camara_on, self.camara_off])
    self.record_methods = it.cycle([self.start_recording,
                                    self.stop_recording])

    self.camara_btn = tk.Button(self, image=camara_img,
                                command=self.camara)
    self.camara_btn.image = camara_img_on
    self.camara_btn.place(relx=pos_x-0.1, rely=pos_y, anchor=tk.CENTER,
                          width=56, height=56)

    self.picture_btn = tk.Button(self, image=picture_img,
                                command=self.take_picture)
    self.picture_btn.image = picture_img
    self.picture_btn.place(relx=pos_x, rely=pos_y, anchor=tk.CENTER,
                          width=56, height=56)

    self.record_btn = tk.Button(self, image=record_img,
                                command=self.record)
    self.record_btn.image = record_img
    self.record_btn.place(relx=pos_x+0.1, rely=pos_y, anchor=tk.CENTER,
                          width=56, height=56)

def camara(self):
    """DEFINE SI LA CÁMARA DEBE ENCENDERSE O APAGARSE"""
    self.camara_btn['image'] = next(self.camaras)
    next(self.camara_methods)()

def camara_on(self):
    """ENCIENDE LA CÁMARA"""

```

```

print('camara encendida')
camera.rotation = 180
camera.start_preview()
camera.preview.fullscreen = False
camera.preview.window = (0, 0, 640, 480)

def camara_off(self):
    """APAGA LA CÁMARA"""
    print('camara apagada')
    camera.stop_preview()

def take_picture(self):
    """TOMA FOTOGRAFÍA"""
    print('taking picture')
    filename = time.strftime("%Y%m%d-%H%M%S")
    camera.capture('/home/pi/Desktop/{ }.jpg'.format(filename))

def record(self):
    """DEFINE SI DEBE REALIZAR GRABACIÓN O PARARLA"""
    self.record_btn['image'] = next(self.record_imgs)
    next(self.record_methods)()

def start_recording(self):
    """REALIZAR GRABACIÓN"""
    print('start recording')
    self.update()
    self.camara_btn.event_generate('<Enter>')
    self.camara_btn.event_generate('<Button-1>')
    self.camara_btn.event_generate('<ButtonRelease-1>')
    self.record_btn.event_generate('<Leave>')
    self.record_btn.event_generate('<Enter>')
    filename = time.strftime("%Y%m%d-%H%M%S")
    camera.start_recording('/home/pi/Desktop/{ }.h264'.format(filename))

def stop_recording(self):
    """PARAR GRABACIÓN"""
    print('stop recording')
    self.update()
    self.camara_btn.event_generate('<Enter>')
    self.camara_btn.event_generate('<Button-1>')
    self.camara_btn.event_generate('<ButtonRelease-1>')
    self.record_btn.event_generate('<Leave>')
    self.record_btn.event_generate('<Enter>')
    camera.stop_recording()

def open_location(self):

```



```

"""VISUALIZACIÓN DE LA POSICIÓN EN GOGGLE MAPS MEDIANTE UN
NAVEGADOR"""

```

```

not_allowed = ('', 'n/a')
if self.latitud.get() not in not_allowed and self.longitud.get()\
not in not_allowed:
    url = 'https://www.google.com/maps/place/{},{ }\
.format(self.latitud.get(), self.longitud.get())
    webbrowser.get('chromium-browser').open_new(url)
else:
    messagebox.showinfo('GPS no listo', 'Los valores del GPS '
        'aÃºn no estÃ¡n listos.')

def read_gps(self):
    """LECTURA DEL GPS"""
    latitud_raw = self.agps_thread.data_stream.lat
    longitud_raw = self.agps_thread.data_stream.lon

    if self.drone.is_moving:
        velocidad = self.agps_thread.data_stream.speed
    else:
        velocidad = 0

    latitud = round(Decimal(latitud_raw), 5) if latitud_raw != 'n/a' else latitud_raw
    longitud = round(Decimal(longitud_raw), 5) if longitud_raw != 'n/a' else
longitud_raw

    self.latitud.set('{} '.format(latitud))
    self.longitud.set('{} '.format(longitud))
    self.velocidad.set('{} km/h'.format(velocidad))

def pump_on(self):
    """ENCENDIDO DE LA BOMBA"""
    if not self.drone.bomba and self.nivel_totalizado < 25:
        self.drone.encender_bomba()
        self.time_start_pump = time.time()
        self.bomba.set('Encendida')
        self.bomba_output.config(fg=Style.DISPLAY_COLOR)

def pump_off(self):
    """APAGADO DE LA BOMBA"""
    if self.drone.bomba:
        self.drone.apagar_bomba()
        self.bomba.set('Apagada')
        self.bomba_output.config(fg=Style.OFF_COLOR)

    # Popup window
    SampleDialog(self, title="Muestra")

```

```
def read_sensors(self):
    """LECTURA DE SENSORES"""
    if not self.serial_conn:
        self.init_serial()
    else:
        temperatura = self.temperatura.get()
        oxigeno = self.oxigeno.get()
        ph = self.ph.get()
        conductividad = self.conductividad.get()

    try:

        while True:
            # se intenta leer un caracter serialmente y decodificarlo
            char = self.ser.read()

            try:
                char = char.decode("utf-8")
            except UnicodeDecodeError:
                break

            # si no se leyÃ³ nada sale del lazo
            if len(char) == 0:
                break

            # si el caracter es un delimiter se lo quita
            if char == '\r':
                char = "

        """
        PARA CUANDO ENCUENTRA UN FIN DE LINEA SE TOMAN LOS
        PARÁMETROS DE LOS SENSORES Y SE LOS DESPLIEGA, CASO
        CONTRARIO SE SIGUE AGREGANDO CARACTERES AL BUFFER.
        """
        if char == '\n':
            self.serial_buffer += "\n"

        datos = str(self.serial_buffer).split(':')

        parametros = [dato.strip() for dato in datos]

        parametros = parametros[1:-1]

        numero_datos = len(parametros)

        # Lectura de parámetros
```

```

temperatura = parametros[0] if numero_datos > 0 and \
    parametros[0] else 0
ph = parametros[1] if numero_datos > 1 and \
    parametros[1] else 0
oxigeno = parametros[2] if numero_datos > 2 and \
    parametros[2] else 0
conductividad = parametros[3] if numero_datos > 3 and \
    parametros[3] else 0
flujo = parametros[4] if numero_datos > 4 and \
    parametros[4] else 0

# Actualizacion del estado de la bomba
if self.drone.bomba:
    self.bomba.set('Encendida')
else:
    self.bomba.set('Apagada')

self.progress_dialog.destroy()

self.temperatura.set('{} Â°C'.format(temperatura))
self.oxigeno.set('{} ppm'.format(oxigeno))
self.ph.set('{} '.format(ph))
self.conductividad.set('{} uS/cm'
    .format(conductividad))

# Creación del dato de nivel por medio del flujo
if self.drone.bomba:
    nivel = float(flujo) * (time.time() - self.time_start_pump)
    self.nivel_totalizado += nivel
    nivel_output = round(
        Decimal(self.nivel_totalizado), 2)
    self.nivel.set('{} ml'.format(
        nivel_output))

    if self.nivel_totalizado > 25 \
        and self.drone.bomba:
        self.pump_off()

self.read_gps()

self.serial_buffer = "" # borrar buffer
else:
    self.serial_buffer += str(char) # añadir al buffer
except SerialException:
    self.serial_conn = False
    self.ser.close()
    self.init_serial()

```



```
# volver a ejecutar función
self.parent.after(500, self.read_sensors)
```

```
-----
import RPi.GPIO as GPIO
class Drone:
    class __Drone:
        def __init__(self):
            """INICIALIZACIÓN DE PINES GPIO PARA CONTROL DE MOTOR Y DE
BOMBA"""
            GPIO.setwarnings(False)
            GPIO.setmode(GPIO.BOARD)
            self.motor_A1 = 29 # color tomate y verde
            self.motor_A2 = 31
            self.motor_B1 = 33 # azul morado
            self.motor_B2 = 35
            self.motor_bomba_1 = 21
            self.motor_bomba_2 = 23
            self.trig = 11
            self.echo = 13
            GPIO.setup(self.motor_A1, GPIO.OUT)
            GPIO.setup(self.motor_B1, GPIO.OUT)
            GPIO.setup(self.motor_A2, GPIO.OUT)
            GPIO.setup(self.motor_B2, GPIO.OUT)
            GPIO.setup(self.motor_bomba_1, GPIO.OUT)
            GPIO.setup(self.motor_bomba_2, GPIO.OUT)
            GPIO.setup(self.trig, GPIO.OUT)
            GPIO.setup(self.echo, GPIO.IN)

        def up(self):
            """MÉTODO PARA QUE EL DRON AVANCE RECTO"""
            print('up')
            GPIO.output(self.motor_A1, GPIO.HIGH)
            GPIO.output(self.motor_A2, GPIO.LOW)
            GPIO.output(self.motor_B1, GPIO.LOW)
            GPIO.output(self.motor_B2, GPIO.HIGH)

        def down(self):
            """MÉTODO PARA RETROCEDER"""
            print('down')
            GPIO.output(self.motor_A1, GPIO.LOW)
            GPIO.output(self.motor_A2, GPIO.HIGH)
            GPIO.output(self.motor_B1, GPIO.HIGH)
            GPIO.output(self.motor_B2, GPIO.LOW)
```

```
def left(self):
    """MÉTODO PARA QUE EL DRON GIRE A LA IZQUIERDA """
    print('left')
    GPIO.output(self.motor_A1, GPIO.LOW)
    GPIO.output(self.motor_A2, GPIO.HIGH)
    GPIO.output(self.motor_B1, GPIO.LOW)
    GPIO.output(self.motor_B2, GPIO.HIGH)

def right(self):
    """MÉTODO PARA QUE EL DRON GIRE A LA DERECHA """
    print('right')
    GPIO.output(self.motor_A1, GPIO.HIGH)
    GPIO.output(self.motor_A2, GPIO.LOW)
    GPIO.output(self.motor_B1, GPIO.HIGH)
    GPIO.output(self.motor_B2, GPIO.LOW)

def stop(self):
    """MÉTODO PARA PARAR EL DRON"""
    print('stop')
    GPIO.output(self.motor_A1, GPIO.LOW)
    GPIO.output(self.motor_A2, GPIO.LOW)
    GPIO.output(self.motor_B1, GPIO.LOW)
    GPIO.output(self.motor_B2, GPIO.LOW)

def encender_bomba(self):
    """MÉTODO PARA ENCENDER LA BOMBA DE SUCCIÓN"""
    print('bomba encendida')
    self.bomba = 'on'

def apagar_bomba(self):
    """MÉTODO PARA APAGAR LA BOMBA DE SUCCIÓN"""
    print('bomba apagada')
    self.bomba = 'off'

@property
def is_moving(self):
    if GPIO.input(self.motor_A1) or GPIO.input(self.motor_A2) or \
        GPIO.input(self.motor_B1) or GPIO.input(self.motor_B2):
        return True
    return False

@property
def bomba(self):
    if GPIO.input(self.motor_bomba_1):
        return True
    return False
```

```
@bomba.setter
def bomba(self, action):
    if action == 'on':
        GPIO.output(self.motor_bomba_1, GPIO.HIGH)
    elif action == 'off':
        GPIO.output(self.motor_bomba_1, GPIO.LOW)

instance = None

def __init__(self):
    if not Drone.instance:
        Drone.instance = Drone.__Drone()
    else:
        pass

def __getattr__(self, name):
    return getattr(self.instance, name)
-----
//MENÚ
# -*- coding: utf-8 -*-
import tkinter as tk
from tkinter import ttk
from models.medicion import Medicion
from models.posicion import Posicion

class MenuNotebook(ttk.Notebook):
    """MENÚ DE LA APLICACIÓN"""
    def __init__(self, parent, *args, **kwargs):
        ttk.Notebook.__init__(self, parent, *args, **kwargs)
        self.parent = parent
        # Logo
        logo_img = tk.PhotoImage(file="~/Documents/usv/imgs/logo.gif",
master=self.parent)

        self.logo = tk.Label(self.parent, image=logo_img)
        self.logo.image = logo_img
        self.logo.place(relx=0.5, y=30, anchor=tk.N)

    def almacenar_medicion(self):
        """
        MÉTODO QUE ALMACENA EL DATO Y LLAMA AL MÉTODO PARA
        ACTUALIZAR TABLA DE MEDICIONES.
        """
        main_tab = self.winfo_children()[0]

        medicion = Medicion(
            temperatura=main_tab.temperatura,
```



```

        oxigeno=main_tab.oxigeno,
        ph=main_tab.ph,
        conductividad=main_tab.conductividad,
    )
    medicion.save()

    data_tab = self.winfo_children()[1]
    data_tab.update_table()

def almacenar_posicion(self):
    """
    MÉTODO QUE ALMACENA EL DATO Y LLAMA AL MÉTODO PARA
    ACTUALIZAR TABLA DE MEDICIONES.
    """
    main_tab = self.winfo_children()[0]
    posicion = Posicion(
        latitud=main_tab.latitud,
        longitud=main_tab.longitud,
    )
    posicion.save()
    data_tab = self.winfo_children()[2]
    data_tab.update_table()
-----
//DATA.PY
# -*- coding: utf-8 -*-
import tkinter as tk
from tkinter import ttk
from styles.main_styles import Style
from models.medicion import Medicion
from models.posicion import Posicion

class DataFrame(tk.Frame):
    """
    PESTAÑA PARA MOSTRAR LOS DATOS ALMACENADOS, BIEN SEA DE
    MEDICIONES O DE POSICIONES ALMACENADAS
    """
    def __init__(self, parent, *args, **kwargs):
        tk.Frame.__init__(self, parent, *args, **kwargs)
        self.parent = parent
        self.name = kwargs['name']
        self.init_frame()

    def init_frame(self):
        """INICIALIZACIÓN DE WIDGETS EN EL FRAME."""
        # Posicionamiento de la tabla
        x_pos = 31.5

```

```

y_pos = 150

self.table = ttk.Treeview(self, height=15)
self.table['show'] = 'headings'

# Table scroll
self.table_scroll = ttk.Scrollbar(self, orient="vertical",
                                   command=self.table.yview)
self.table_scroll.place(x=x_pos+622, y=y_pos, height=318)
self.table.configure(yscrollcommand=self.table_scroll.set)

if self.name == 'mediciones':

    header = 'Mediciones realizadas'

    self.table['columns'] = ['id', 'temperatura', 'oxigeno', 'ph',
                             'conductividad', 'fecha']

    self.table.column('id', width=70)
    self.table.column('temperatura', width=100)
    self.table.column('oxigeno', width=130)
    self.table.column('ph', width=50)
    self.table.column('conductividad', width=120)
    self.table.column('fecha', width=150)

    self.table.heading('id', text='No.')
    self.table.heading('temperatura', text='Temperatura')
    self.table.heading('oxigeno', text='Oxígeno Disuelto')
    self.table.heading('ph', text='pH')
    self.table.heading('conductividad', text='Conductividad')
    self.table.heading('fecha', text='Fecha y Hora')

elif self.name == 'posiciones':

    header = 'Localizaciones almacenadas'

    self.table['columns'] = ['id', 'latitud', 'longitud', 'fecha']

    self.table.column('id', width=71)
    self.table.column('latitud', width=183)
    self.table.column('longitud', width=183)
    self.table.column('fecha', width=183)

    self.table.heading('id', text='No.')
    self.table.heading('latitud', text='Latitud')
    self.table.heading('longitud', text='Longitud')
    self.table.heading('fecha', text='Fecha y Hora')

```

```

else:
    raise ValueError(
        "The DataFrame's name must be either 'mediciones' "
        " or 'posiciones'.")

# Header
self.label_header = tk.Label(self, text=header,
                              font=Style.HEADER_FONT)
self.label_header.place(relx=0.5, rely=0.15, anchor=tk.N)

# Se quita el id y se añade el número de valor en la base de datos
datos = self.get_datos()

self.show_data(datos)
self.table.place(x=x_pos, y=y_pos)

# Botón de borrado de datos
delete_button = tk.Button(self, text='Borrar {}'.format(self.name),
                           font=Style.TEXT_FONT,
                           command=self.delete_data)
delete_button.place(x=x_pos+479, y=y_pos+350, width=158)

def update_table(self):
    """MÉTODO PARA ACTUALIZAR TABLA DE DATOS"""
    old_data = self.table.get_children()
    for data in old_data:
        self.table.delete(data)
    datos = self.get_datos()
    self.show_data(datos)

def show_data(self, datos):
    """MÉTODO PARA MOSTRAR TABLA DE DATOS"""
    for idx, dato in enumerate(datos):
        self.table.insert("", idx, values=dato)

def get_datos(self):
    """MÉTODO PARA OBTENER LAS DATOS DE LA BASE DE DATOS"""
    if self.name == 'mediciones':
        datos = [(idx+1,) + dato[1:] for idx, dato in
                  enumerate(Medicion.all())]
    elif self.name == 'posiciones':
        datos = [(idx+1,) + dato[1:] for idx, dato in
                  enumerate(Posicion.all())]
    return datos

def delete_data(self):

```



```

        """MÉTODO PARA BORRAR LOS DATOS"""
        if self.name == 'mediciones':
            Medicion.clear()
        elif self.name == 'posiciones':
            Posicion.clear()
        self.update_table()

-----
//DATABASE.PY
# -*- coding: utf-8 -*-
import sqlite3

class Database(object):
    """CLASE QUE SE ENCARGA DEL MANEJO DE LA BASE DE DATOS, CREA
    LAS TABLAS Y LA CONEXIÓN
    """
    @classmethod
    def connect(cls):
        """CREA LA CONEXIÓN A LA BASE DE DATOS"""
        try:
            db = sqlite3.connect('/home/pi/Documents/usv/db.sqlite3')
        except sqlite3.Error as e:
            print("Database error: %s" % e)
        except Exception as e:
            print("Exception in _query: %s" % e)
        return db

    @classmethod
    def create_tables(cls):
        """CREA LAS TABLAS NECESARIAS PARA EL ALMACENAMIENTO"""
        create_mediciones_table = """
        CREATE TABLE IF NOT EXISTS mediciones (
            IdMedicion INTEGER PRIMARY KEY,
            Temperatura FLOAT(4,2) NOT NULL,
            Oxigeno FLOAT(4,2) NOT NULL,
            pH FLOAT(4,2) NOT NULL,
            Conductividad FLOAT(4,2) NOT NULL,
            Fecha DATETIME DEFAULT CURRENT_TIMESTAMP
        );
        """
        create_posiciones_table = """
        CREATE TABLE IF NOT EXISTS posiciones (
            IdPosicion INTEGER PRIMARY KEY,
            Latitud FLOAT(4,2) NOT NULL,
            Longitud FLOAT(4,2) NOT NULL,
            Fecha DATETIME DEFAULT CURRENT_TIMESTAMP
        );
        """

```

```

db = cls.connect()
cursor = db.cursor()
cursor.execute(create_mediciones_table)
cursor.execute(create_posiciones_table)
db.commit()
db.close()

-----
//MEDICION.PY
# -*- coding: utf-8 -*-
from .database import Database
from datetime import datetime
from tkinter import messagebox
from .model import Model

class Medicion(Model):
    table_name = 'mediciones'

    def __init__(self, *args, **kwargs):
        self.temperatura = kwargs['temperatura'].get().split()[0]
        self.oxigeno = kwargs['oxigeno'].get().split()[0]
        self.ph = kwargs['ph'].get()
        self.conductividad = kwargs['conductividad'].get().split()[0]

    def save(self):
        """ALMACENADO DE MEDICIONES"""
        db = Database.connect()
        cursor = db.cursor()
        medidas = (self.temperatura, self.oxigeno, self.ph,
                    self.conductividad, datetime.now())
        cursor.execute("INSERT INTO mediciones (
                        Temperatura, Oxigeno, pH, Conductividad,
                        Fecha) VALUES (?, ?, ?, ?, ?)
                        ", medidas)
        db.commit()
        db.close()
        messagebox.showinfo('Medición almacenada!',
                            'La medición ha sido almacenada exitosamente.')

-----
//MODEL.PY
# -*- coding: utf-8 -*-
from .database import Database

class Model(object):

    table_name = "

```

```
@classmethod
def all(cls):
    """OBTENCIÓN DE TODOS LOS DATOS."""
    db = Database.connect()
    cursor = db.cursor()
    cursor.execute('SELECT * FROM {}'.format(cls.table_name))
    data = cursor.fetchall()
    db.close()
    return data

@classmethod
def clear(cls):
    """BORRADO DE TODAS LAS ENTRADAS."""
    db = Database.connect()
    cursor = db.cursor()
    cursor.execute('DELETE FROM {}'.format(cls.table_name))
    db.commit()
    db.close()

-----
//POSICION.PY
# -*- coding: utf-8 -*-
from .database import Database
from datetime import datetime
from tkinter import messagebox
from .model import Model

class Posicion(Model):

    table_name = 'posiciones'

    def __init__(self, *args, **kwargs):
        self.latitud = kwargs['latitud']
        self.longitud = kwargs['longitud']

    def save(self):
        """ALMACENADO DE POSICIONES"""
        db = Database.connect()
        cursor = db.cursor()
        posiciones = (self.latitud.get(), self.longitud.get(), datetime.now())
        cursor.execute("INSERT INTO posiciones (
            Latitud, Longitud,
            Fecha) VALUES (?, ?, ?)
            ", posiciones)
        db.commit()
        db.close()
        messagebox.showinfo('Posición almacenada!',
            'La posición ha sido almacenada exitosamente.')
```



```

-----
//DIALOG.PY
import tkinter as tk

class Dialog(tk.Toplevel):

    def __init__(self, parent, title=None, **kwargs):
        tk.Toplevel.__init__(self, master=parent, **kwargs)
        self.transient(parent)

        if title:
            self.title(title)

        self.parent = parent

        self.result = None

        body = tk.Frame(self)
        self.initial_focus = self.body()
        body.pack(padx=5, pady=5)

        self.set_geometry()

        self.buttonbox()

        self.grab_set()

        if not self.initial_focus:
            self.initial_focus = self

        self.protocol('WM_DELETE_WINDOW', self.close)

        self.initial_focus.focus_set()

        self.wait_window(self)

# construction hooks

    def set_geometry(self):
        self.parent.update()
        self.geometry('+{ }+{ }'.format(self.parent.winfo_rootx()+200,
                                         self.parent.winfo_rooty()+200))

    def body(self):
        """Create dialog body. Return widget that should have
        initial focus.  this method should be overridden"""
        pass

```

```
def buttonbox(self):
    pass

def close(self, event=None):
    # put focus back to the parent window
    self.parent.focus_set()
    self.destroy()

-----
//SAMPLE_DIALOG.PY
import tkinter as tk
from .dialog import Dialog
from styles.main_styles import Style

class SampleDialog(Dialog):
    def body(self):
        glass_img = tk.PhotoImage(file="~/Documents/usv/imgs/glass.png", master=self)
        glass = tk.Label(self, image=glass_img)
        glass.image = glass_img
        glass.pack()

        info_message = 'Se han recogido { } de agua'.format(
            self.parent.nivel.get())
        message_label = tk.Label(
            self, text=info_message,
            font=Style.POPUP_FONT)
        message_label.pack(padx=10)

    def buttonbox(self):
        btn_box = tk.Frame(self)

        if self.parent.nivel_totalizado < 25:
            ok_btn = tk.Button(btn_box, text='Ok', width=10,
                               command=self.reset_level,
                               font=Style.TEXT_FONT)
            ok_btn.pack(side=tk.LEFT, padx=5, pady=5)
        else:
            advertencia = 'ADVERTENCIA: Retire el vaso antes de continuar \'
            \'tomando muestras.'
            advertencia_label = tk.Label(
                self,
                text=advertencia,
                font=Style.TEXT_FONT
            )
            advertencia_label.pack(padx=10, pady=5)
            ok_btn = tk.Button(btn_box, text='Ok', width=10,
                               command=self.reset_level,
                               font=Style.TEXT_FONT)
```

```

ok_btn.pack(side=tk.LEFT, padx=5, pady=5)
self.geometry('+{ }+{ }'.format(self.parent.winfo_rootx()+100,
                                self.parent.winfo_rooty()+100))

self.bind('<Return>', self.reset_level)
self.bind('<Escape>', self.close)

btn_box.pack()

def reset_level(self):
    self.parent.time_start_pump = 0.0
    self.parent.nivel_totalizado = 0.0
    self.parent.nivel.set('0 ml')
    self.close()
-----
//MAIN_STYLES.PY
# -*- coding: utf-8 -*-
class Style:
    """COLORES Y ESTILOS GENERALES PARA LA APLICACIÓN"""
    TEXT_FONT = ('Helvetica', 12)
    POPUP_FONT = ('Helvetica', 12, 'bold')
    HEADER_FONT = ('Helvetica', 16)
    WHITE = "#fff"
    BLACK = "#000"
    OFF_COLOR = "#929292"
    DISPLAY_COLOR = "#0f0"
    BACKGROUND_COLOR = "#ecf0f1"
    PRIMARY_COLOR = "#2c3e50"
    SECONDARY_COLOR = "#8e44ad"

```


ANEXO 05. MEDIDOR DE OXÍGENO PINPOINT II

American Marine is proud to introduce **PINPOINT® II Dissolved Oxygen Monitor**. The only truly affordable and accurate DO monitoring system. Powered by a 9-volt battery (not included). **PINPOINT® II** will monitor for approximately 150 total hours before battery replacement is visually indicated on the display as "LOBAT". A 9-volt battery should be utilized for field or intermittent use. For continuous monitoring, the **PINPOINT® AC Adapter Kit** should be installed. The battery cover is located on the back of the meter toward the bottom (diagram #3-7).

NOTE: The Oxygen Monitor unit is NOT waterproof and must be operated on a dry surface. Liquid contact on the printed circuit board will cause corrosion and void warranty.

•Set-up

- 1) Slide the bottom switch #3-3 to the CAL position (toward the right).
- 2) With the probe NOT CONNECTED turn the zero adjust screw #3-4 until the display reads 00.0
- 3) Fill Membrane Cap with Electrolyte Fluid and screw onto probe body (finger tight only). Remove the dark blue silicone rubber cap at the end of the Membrane Cap.
- 4) Connect the probe to the meter by carefully lining up the pins on the plug to the input on the meter. Be sure to fully insert the connector.
- 5) Wait several minutes for the probe to polarize and stabilize on the meter.
- 6) Turn the O2 calibration screw #3-5 so that the display will read 20.9. This is the percentage of Oxygen in our atmosphere and is widely considered to be a highly accurate calibration procedure.
- 7) Slide the bottom switch #3-3 to the DO position (toward the left). The monitor is now ready for use.

•Replacement of the membrane cap and liquid electrolyte

The **PINPOINT II Oxygen Monitor®** utilizes a very high performance polarographic electrode. This Membrane Cap at the end of the probe should be changed frequently to ensure optimum performance. Additional Membrane Caps are available. Please refer to the probe diagram below:



- 1) Unscrew the Membrane Cap as from the probe body. This is 1-3/8" from the end of the probe.
- 2) Fill a new Membrane Cap with electrolyte fluid. Screw the Membrane Cap onto the probe body (finger tight). A small amount of electrolyte fluid may leak out.
- 3) A dark blue protective silicone rubber cap is installed over the tip of the Membrane Cap. This will protect the clear Teflon Membrane surface during storage. This dark blue cap must be removed for both calibration and sample measurement. Reinstall the dark blue cap during storage.

Please read "Important Points to Remember" printed on back

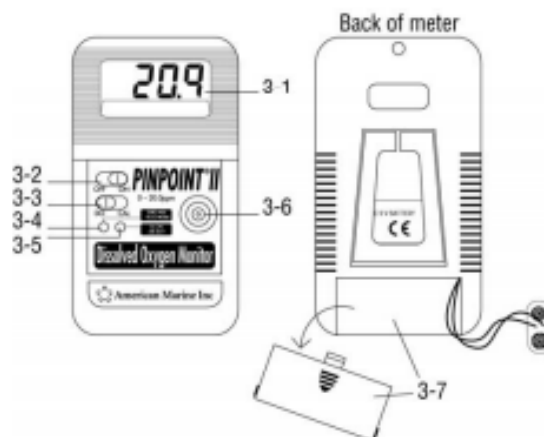
PINPOINT® II

High-Performance Dissolved Oxygen Monitor

User's Guide

American Marine Inc.
54 Danbury Road, Suite 172
Ridgefield, CT 06877 U.S.A.
Phone/FAX 203-205-0811
www.americanmarineusa.com

©Copyright 1998
American Marine Inc.



Also available:

- PINPOINT® ORP Monitor (REDOX)
- PINPOINT Wireless Thermometer
- Replacement PINPOINT® pH, ORP, Oxygen electrodes
- High Precision pH Calibration Fluids 4.00, 7.00, 10.00
- PINPOINT® Salinity Monitor
- PINPOINT® pH Monitor
- Teflon Membrane Caps and Electrolyte Fluid
- PINPOINT® Conductivity Monitor (Freshwater Hardness)
- 110V Adapter Kit (includes modified battery compartment door)
- PINPOINT pH Controller
- PINPOINT ORP (REDOX) Controller
- PINPOINT Calcium Monitor

Important Points to Remember

- **PINPOINT II Oxygen Monitor** utilizes a high performance polarographic probe. This probe is epoxy sealed and can be total submerged if desired.
- **PINPOINT II Oxygen Meter** is NOT waterproof and must be operated on a dry surface. Liquid contact with the printed circuit board will cause corrosion and void warranty.
- Always wait a minute or two for the automatic temperature compensation to take place before recording any readings.
- Probe should be inserted about 1 inch into the liquid to be measured. Probe may be totally submerged if desired.
- While taking a reading you should "swirl" the probe or confirm that there is adequate flow in the sample fluid to insure an accurate reading.
- The clear Teflon Membrane tip should be moistened with fresh water when taking readings in air and during the regular calibration procedure.
- It is important that the probe be given a few minutes to polarize during the calibration procedure. This helps to stabilize the readings.
- The **PINPOINT II** polarographic oxygen probe contains an internal temperature thermistor and is automatically temperature compensated for the range of 32°F to 122°F (0° to 50°C). Therefore the displayed value is already compensated and correct.
- The silver metal sleeve located on the probe tip under the membrane cap should be gently cleaned with steel wool every few months to prevent glazing

Warranty

PINPOINT II Oxygen Monitor® by American Marine Inc. is warranted to be free of defects in material and workmanship for a period of 2 years from date of sale. Positive proof of purchase is required for warranty claim.

Removal or alteration of the serial number will void warranty.

American Marine Inc. will not be liable for any costs of removal, installation, transportation charges, or any other charges, which may result in connection with a warranty claim.

American Marine Inc. will not be liable for any damage or wear to products or livestock caused by abnormal operating conditions, water damage, abuse, misuse, unauthorized alternation or repair or if the product was not installed in accordance with the printed operating instructions.

Any defective product to must be sent freight prepaid with appropriate documentation supporting the warranty claim. Replacement or repair will be at the discretion of American Marine Inc. Typical turnaround time within 48 hours. Overnight delivery available. American Marine Inc.

54 Danbury Road, Suite 172
Ridgefield, CT 06877 U.S.A.
www.americanmarineusa.com

ANEXO 06. MEDIDOR DE CONDUCTIVIDAD ELÉCTRICA VA 1022

www.vimael.com/index.php/medidores-en-liquid...



Conductimetro Medidor de conductividad tds y temperatura marca Va modelo 1022 con sensor interno

Detalles

Categoría: Medidores en líquidos ph tds orp conductividad temperatura y otros



CONDUCTIVIDAD TDS Y TEMPERATURA MARCA VA 1022 CON SENSOR INTERNO

Este equipo es adecuado para medir la conductividad, tds y temperatura en líquidos, viene con sensor interno,

CARACTERÍSTICAS:

Rango de conductividad: 0 – 1999 us/cm

Resolución de conductividad: 1 us/cm

Precisión de conductividad: + - 2%

Rango de tds: 0 a 1999 ppm

Rango de temperatura: 0 a 70 grados centígrados

ATC con compensación de temperatura de 0-70 grados centígrados

Material del equipo ABS IP65

Tecla de retención de datos

Alimentación 2 pilas AAA

Use agua destilada para limpieza de la punta de prueba, seque bien y ponga el cobertor

INCLUYE:

01 Conductimetro marca VA modelo 1022

01 Bolsa de portar

ANEXO 07. SENSOR DE PH Y TEMPERATURA

<https://www.diyamore.cc/products/diyamore-liquid-ph-v...>

Description Download Delivery Payment Review

Item Special :

- Heating voltage: $5 \pm 0.2V$ (AC -- DC)
- Working current: 5-10mA
- The detection concentration range: PH0-14
- The detection range of temperature: 0-60 centigrade
- The response time: $\leq 5S$
- Stability time: $\leq 60S$
- Power consumption: $\leq 0.5W$
- The working temperature: -10~50 centigrade (the nominal temperature 20 centigrade)
- Working humidity: 95%RH (nominal humidity 65%RH)
- Service life: 3 years
- Size: 42mm x 32mm x 20mm
- Weight: 25g
- The output: analog voltage signal output

Package Include:

1 x Liquid PH Value Detection Sensor Module Monitoring Control Module



ANEXO 08. SENSOR DE FLUJO

<https://naylampmechatronics.com/sensores-liquido/10...>



Sensor de flujo de agua 1/2\"
YF-S201

Modelo SEN-FLOW-050

El sensor de flujo permite medir el caudal de agua en una tubería y así poder conocer el consumo.

Imprimir

S/. 20,00

Cantidad
1

Añadir al carrito

Añadir a la lista de deseos

DESCRIPCIÓN

INFO

Un sensor de flujo o caudalímetro es un instrumento para la medición de caudal o gasto volumétrico de un fluido. El caudal es la cantidad de líquido o fluido (volumen) que circula a través de una tubería por unidad de tiempo, por lo general se expresa en: litros por minutos (l/m), litros por hora (l/h), metros cúbicos por hora (m³/h), etc.). Los caudalímetros suelen colocarse directamente en la tubería que transporta el fluido. También suelen llamarse medidores/sensores de caudal, medidores de flujo o flujómetros.

El sensor de flujo de agua de 1/2" YF-S201 sirve para medir caudal de agua en tuberías de 1/2" de diámetro. También puede ser empleado con otros líquidos de baja viscosidad, como: bebidas gasificadas, bebidas alcohólicas, combustible, etc. Es un caudalímetro electrónico de tipo turbina. Compatible con sistemas digitales como Arduino, PIC, Raspberry PI, PLCs. El sensor posee tres cables: rojo (VCC: 5VDC), negro (tierra) y amarillo (salida de pulsos del sensor de efecto Hall). Con la ayuda de este sensor podrás ingresar al mundo de la Domótica, monitoreando el consumo de agua en tu hogar, o puedes hacer un dispensador de volumen automatizado con la ayuda de una válvula adicional.

El funcionamiento del sensor es de la siguiente forma: el caudal de agua ingresa al sensor y hace girar una turbina, la turbina está unida a un imán que activa un sensor de efecto Hall, que a su vez emite un pulso eléctrico que puede ser leído por la entrada digital de un Arduino o PLC. El sensor de efecto Hall está aislado del agua, de manera que siempre se mantiene seco y seguro. Como el volumen de agua por cada pulso es fijo y de un valor conocido (promedio) podemos contar la cantidad de pulsos por unidad de tiempo (segundo o minuto), luego multiplicar el valor de volumen/pulso por la cantidad de pulsos y así determinar el caudal o flujo de agua. Se recomienda utilizar interrupciones por hardware en el Arduino para detectar o contar los pulsos del sensor. Tenga en cuenta que este no es un sensor de precisión por lo que la orientación, presión del agua y otras condiciones pueden afectar la medición. Se recomienda calibrar el sensor realizando mediciones con volúmenes conocidos. Calibrado puede llegar a tener una precisión de hasta 10%.

FÓRMULA: Flujo del agua en L/min = Pulsaciones del sensor (Hz) / 7.5

ESPECIFICACIONES TÉCNICAS


- Modelo: YF-S201
- Voltaje de operación: 5V - 18V DC
- Consumo de corriente: 15mA (5V)
- Capacidad de carga: 10mA (5 VDC)
- Salida: Onda cuadrada pulsante
- Rango de Flujo: 1-30L/min
- Volumen promedio por pulso: 2.25mL
- Pulsos por litro: 450
- Factor de conversión: 7.5
- Rosca externa: 1/2" NPS
- Presión de trabajo máx.: 1.75MPa (17 bar)
- Temperatura de funcionamiento: -25°C a 80°C
- Material: Plástico color negro

ANEXO 09. PASOS DE INSTALACIÓN DE LA PLATAFORMA ARDUINO

- Ingresar a la web con la siguiente dirección electrónica: <https://www.arduino.cc/>
- Ingresar a software y dar clic en descargas, a continuación, se muestra la página para descargar el software Arduino para Windows, Mac o Linux.



Descarga el IDE de Arduino



ARDUINO 1.8.9

El software de código abierto Arduino (IDE) facilita escribir código y cargarlo en la pizarra. Se ejecuta en Windows, Mac OS X y Linux. El entorno está escrito en Java y se basa en Procesamiento y otro software de código abierto. Este software se puede utilizar con cualquier placa Arduino. Consulte la Introducción página para obtener instrucciones de instalación.

Windows Installer, para Windows XP y hasta el archivo ZIP de **Windows** para una instalación no administrativa

La aplicación de Windows requiere Win 8.1 o 10 [Get Windows](#)

Mac OS X 10.8 Mountain Lion o más nuevo

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Notas de la versión de las fuentes de verificación del código fuente (sha512)

ANEXO 10. PASOS DE INSTALACIÓN DE LA PLATAFORMA RASPBERRY

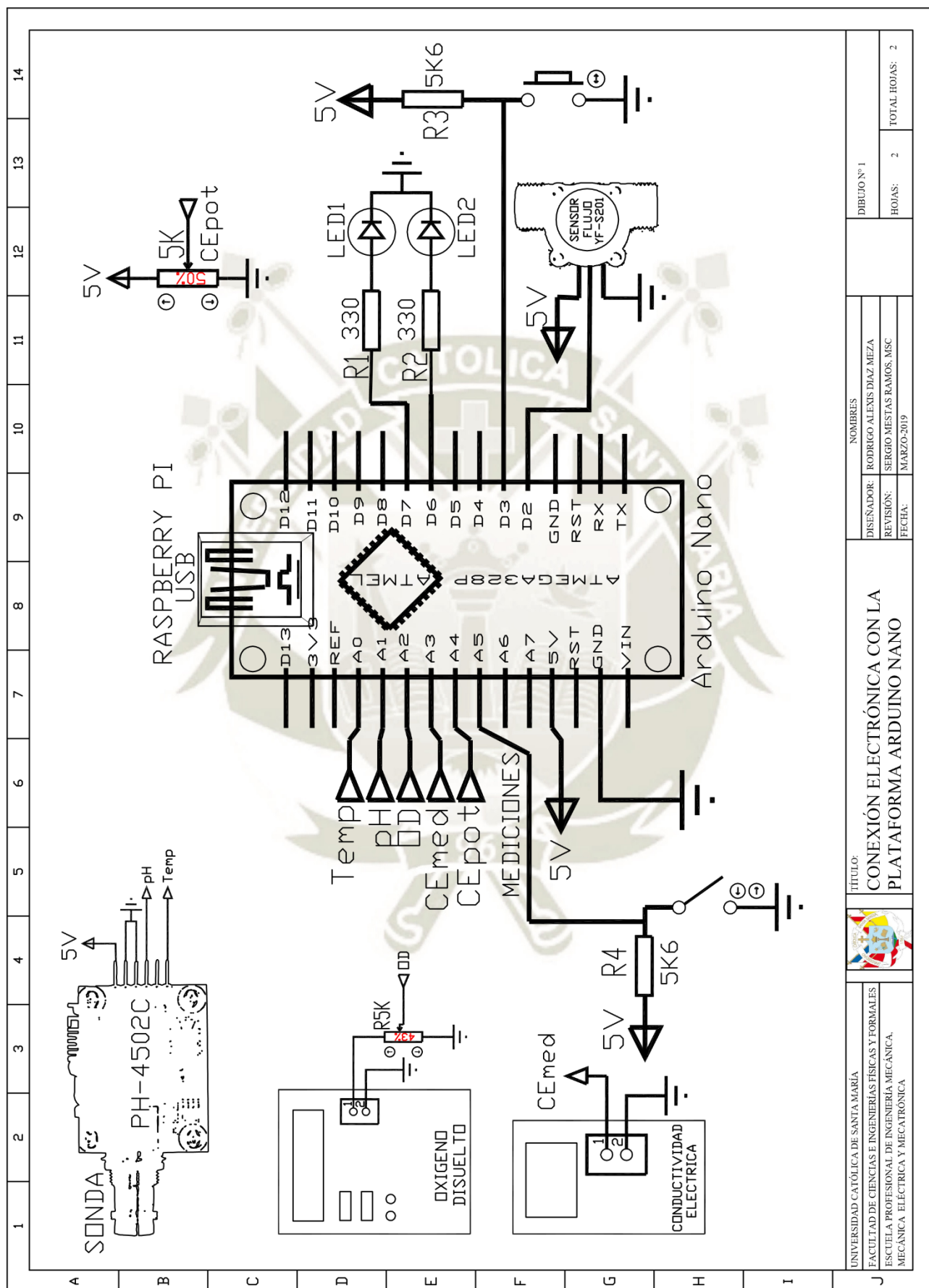
- Ingresar a la web con la siguiente dirección electrónica: <https://www.raspberrypi.org/>
- Ingresar a software y dar clic en descargas, a continuación, se muestra la página para descargar el software Raspbian para Windows o Mac.

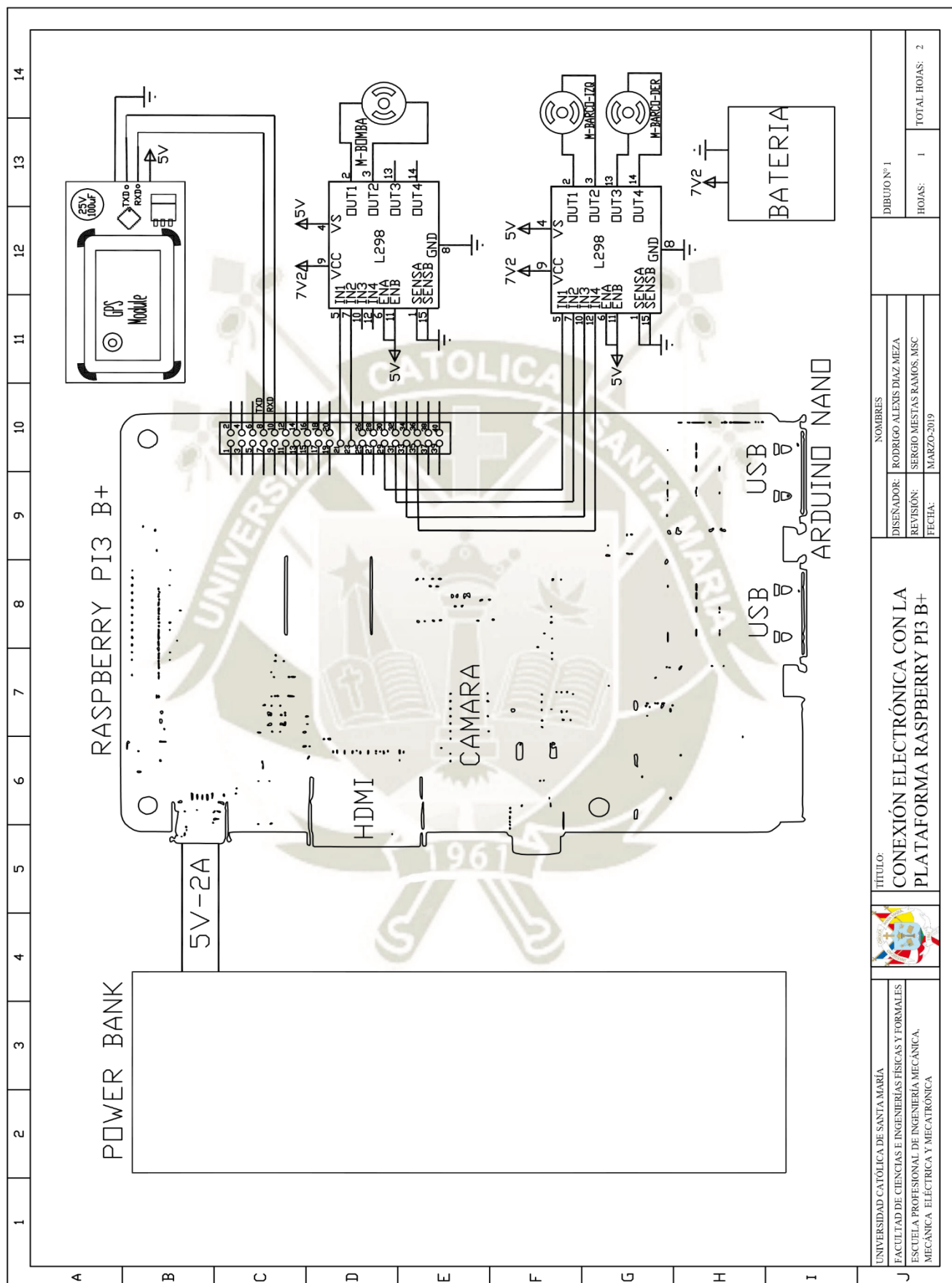


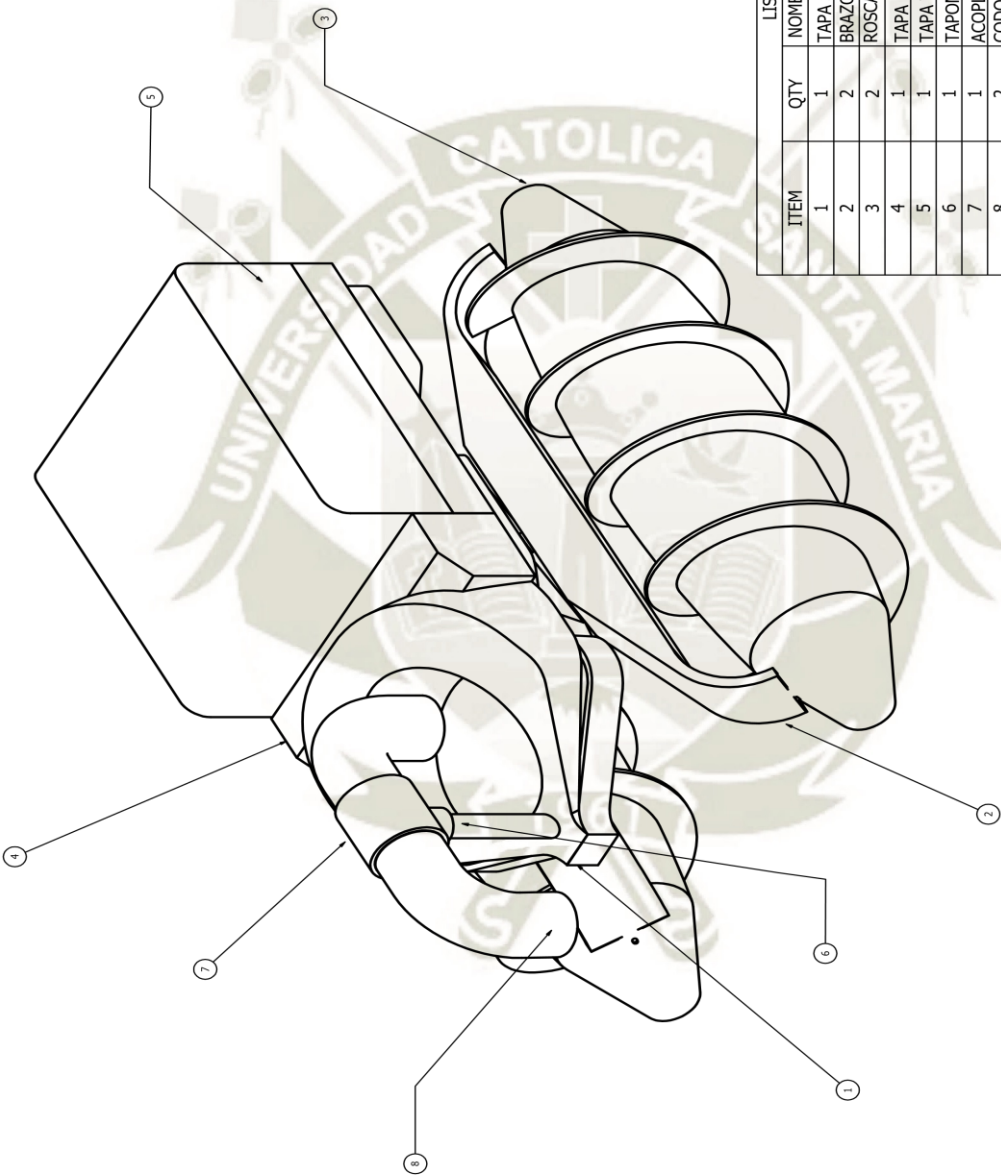
ANEXO 11. COSTO DEL VEHÍCULO DE SUPERFICIE NO TRIPULADO PARA MEDIR LA CALIDAD DEL AGUA

MATERIALES				
#	Cant.	Rubro	Precio	Total
1	1	Vehículo flotador (barco)	\$ 67,20	\$ 67,20
2	1	Kit Raspberry Pi	\$ 89,99	\$ 89,99
3	1	Cámara para la Raspberry	\$ 24,95	\$ 24,95
4	1	Micro SD Sony 32GB	\$ 20,22	\$ 20,22
5	1	Cable USB para IP	\$ 1,59	\$ 1,59
6	1	Módulo GPS	\$ 15,00	\$ 15,00
7	1	Sensor de Conductividad	\$ 55,00	\$ 55,00
8	1	Sensor de Oxígeno Disuelto	\$ 328,78	\$ 328,78
9	1	Sensor de pH	\$ 50,00	\$ 50,00
10	1	Cargador Portátil (POWER BANK)	\$ 20,00	\$ 20,00
11	2	Placas L298 (para control de motores)	\$ 2,00	\$ 4,00
12	1	Sensor de flujo	\$ 7,25	\$ 7,25
13	1	Sensor Infrarrojo	\$ 4,00	\$ 4,00
14	1	Nano Arduino + Cable	\$ 7,00	\$ 7,00
15	1	Batería 9V	\$ 4,00	\$ 4,00
16	1	Caja plástica transparente y canaletas	\$ 4,00	\$ 4,00
17	1	Mini bomba (motor dc + aletas + caja)	\$ 5,00	\$ 5,00
18	2	Codos 1/2 "	\$ 0,35	\$ 0,70
19	2	Pitón 1/2 "	\$ 3,50	\$ 7,00
20	2	Arandelas	\$ 0,16	\$ 0,32
21	1	Manguera 5/16 "	\$ 0,30	\$ 0,30
22	1	Tapón y Canaleta	\$ 0,20	\$ 0,20
23	1	Placa PCB para sensores	\$ 15,00	\$ 15,00
24	1	Caja plástica	\$ 4,00	\$ 4,00
25	1	Baquelita 5x10 cm	\$ 4,25	\$ 4,25
26	1	Pulsador	\$ 0,02	\$ 0,02
27	2	Leds	\$ 0,05	\$ 0,10
28	12	Molex de 2 pines	\$ 0,34	\$ 4,08
29	2	Molex de 3 pines	\$ 0,39	\$ 0,78
30	2	Molex de 4 pines	\$ 0,45	\$ 0,90
31	1	Molex de 5 pines	\$ 0,55	\$ 0,55
32	1	Molex de 10 pines	\$ 0,85	\$ 0,80
33	1	Espadín hembra	\$ 0,13	\$ 0,13
34	1	Cable UTP (1 metro)	\$ 0,55	\$ 0,55
35	3	Resistencias (1/4 w)	\$ 0,02	\$ 0,06
36	1	Potenciómetro de precisión (5k)	\$ 0,53	\$ 0,53
37	1	Potenciómetro logarítmico (5k)	\$ 0,25	\$ 0,25


38	1	Funda de ácido (para las placas PCB)	\$ 0,35	\$ 0,35
39	1	Papel termotransferible e impresión	\$ 2,00	\$ 2,00
SUBTOTAL				\$ 750,85
MANO DE OBRA				
40	Diagnóstico, diseño y elaboración de placa para sensores		\$ 85,00	\$ 85,00
41	Diseño e implementación de mini bomba para succión de agua		\$ 15,00	\$ 15,00
42	Costo del software para la plataforma Arduino Nano		\$ 150,00	\$ 150,00
43	Costo del software para la plataforma Raspberry Pi		\$ 500,00	\$ 500,00
44	Implementación de los componentes electrónicos		\$ 50,00	\$ 50,00
SUBTOTAL				\$ 800,00
TOTAL				\$ 1550,85

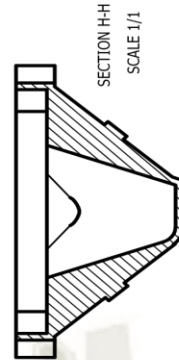





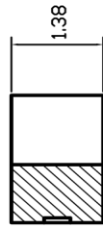
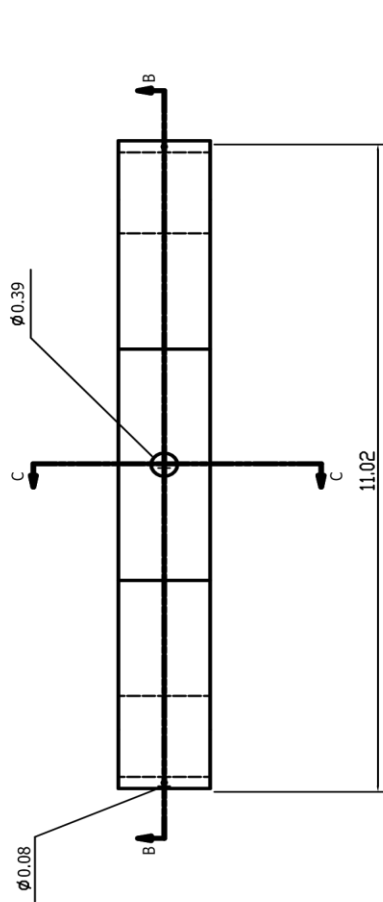


LISTA DE COMPONENTES			
ITEM	QTY	NOMBRE DE COMPONENTE	DESCRIPCION
1	1	TAPA SUPERIOR	HDPE - MANUFACTURADO
2	2	BRAZO	HDPE - MANUFACTURADO
3	2	ROSCA	HDPE - MANUFACTURADO
4	1	TAPA DELANTERA	RESINA - MANUFACTURADO
5	1	TAPA TRASERA	RESINA - MANUFACTURADO
6	1	TAPON	RESINA - MANUFACTURADO
7	1	ACOPLE	PVC - MANUFACTURADO
8	2	CODO	PVC - MANUFACTURADO

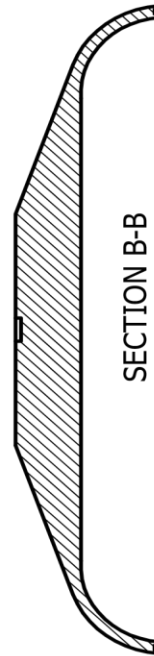
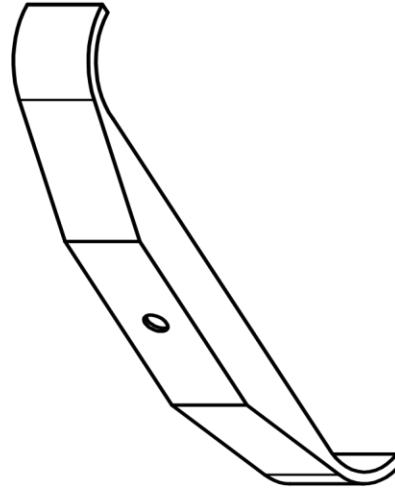
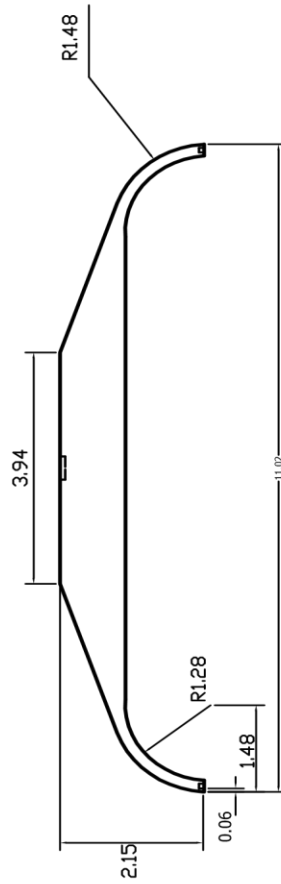
	UNIVERSIDAD CATÓLICA DE SANTA MARÍA Facultad de Ciencias e Ingenierías Físicas y Formales Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	PLANO DE ENSAMBLE: VEHÍCULO DE SUPERFICIE NO TRIPULADO USV	DISEÑADO POR: DIBUJADO POR: REVISION:	NOMBRE Rodrigo Alexis Diaz Meza Rodrigo Alexis Diaz Meza	FECHA	CÓDIGO DE DIBUJO USV-001RADM	REVISION
					MAR - 2019		1.0
					MAR - 2019		A4
					HOJA		1 DE 9




	UNIVERSIDAD CATÁLICA DE SANTA MARÍA	PLANO DE COMPONENTE:		NOMBRE		FECHA	CÓDIGO DE DIBUJO USV-002RADM	REVISION 1.0
	Facultad de Ciencias e Ingenierías Físicas y Formales	TAPA SUPERIOR		DISEÑADO POR	Rodrigo Alexis Díaz Meza	MAR - 2019		
	DIBUJADO POR			Rodrigo Alexis Díaz Meza	MAR - 2019	ESCALA: 1:1		
	REVISIÓN:					HOJA 2 DE 9		
CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos.		A4						

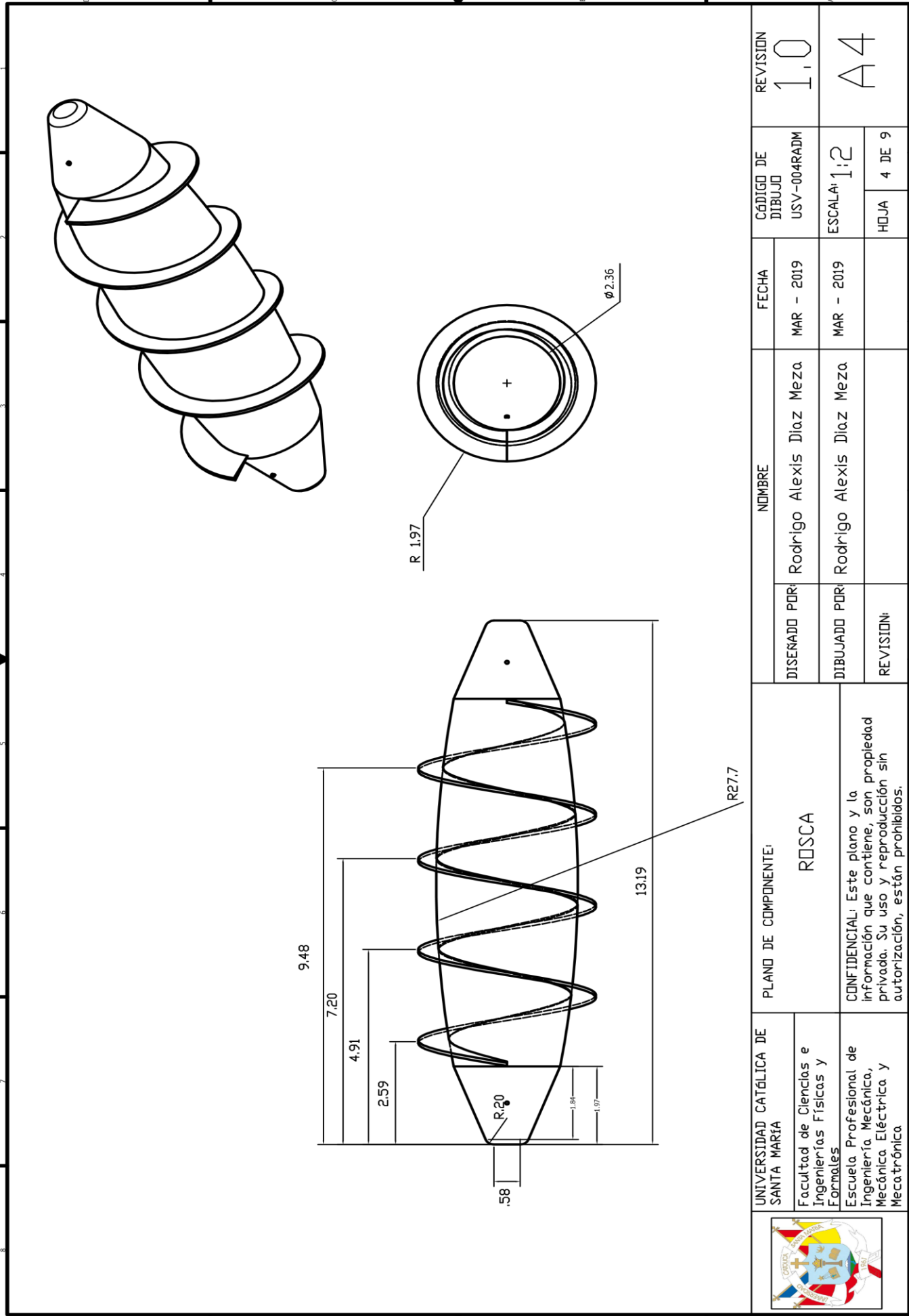



SECTION C-C
SCALE 1 : 1

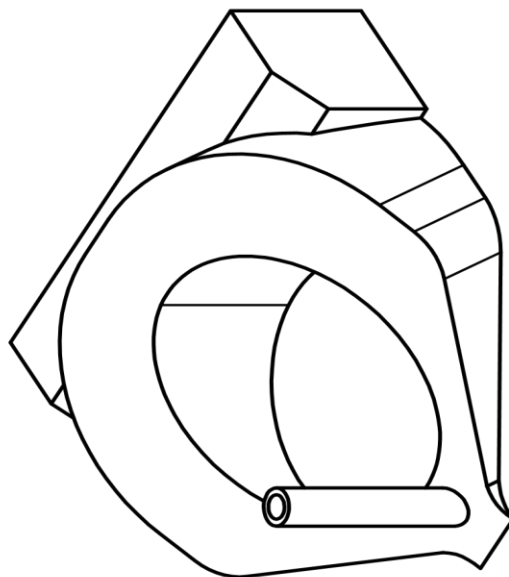
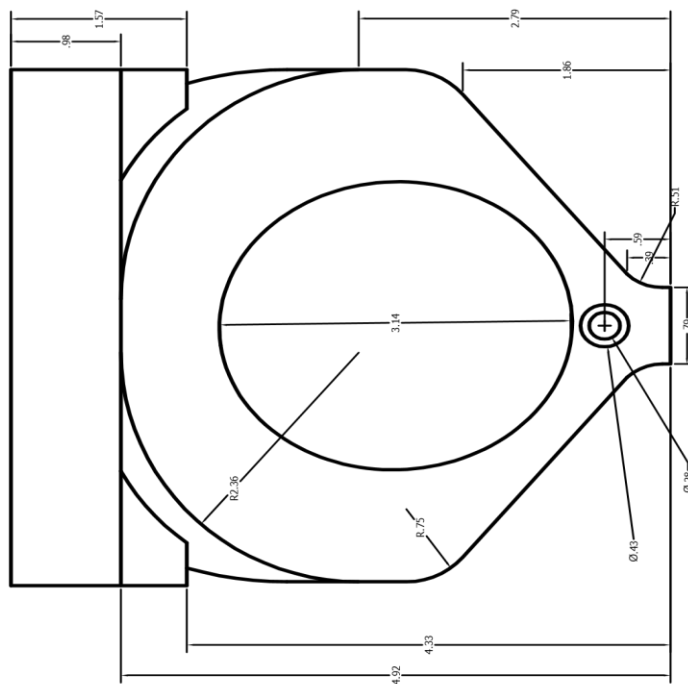
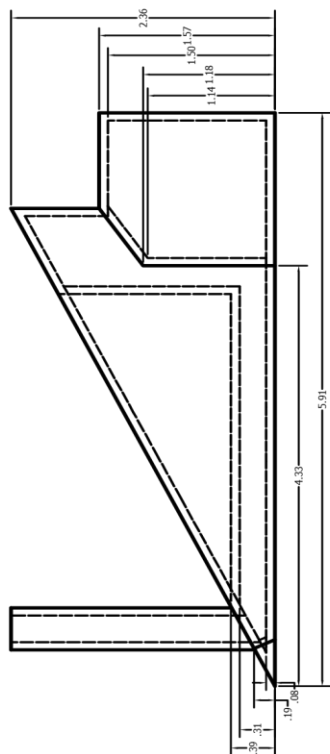



SECTION B-B
SCALE 1 : 1

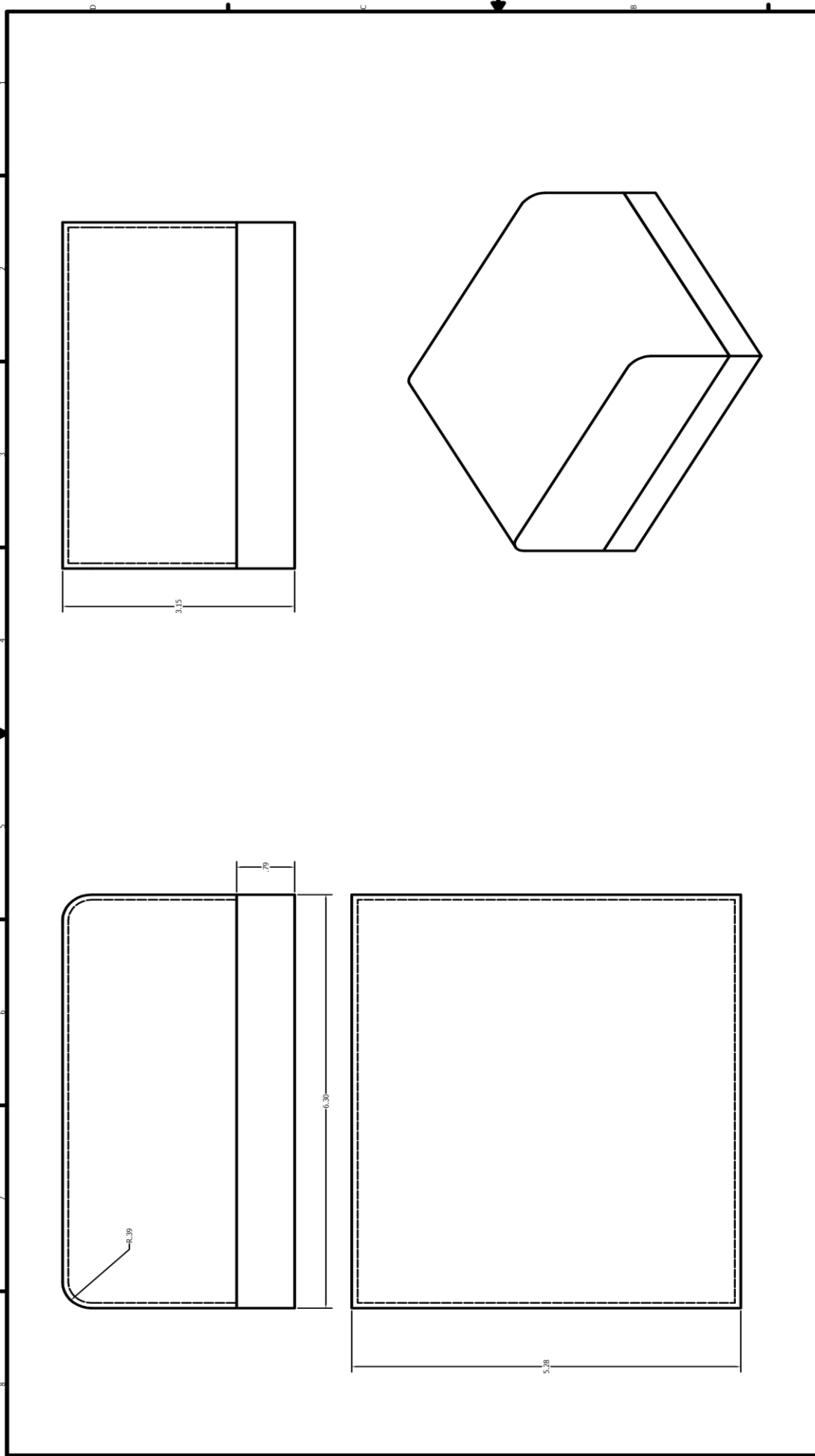
	UNIVERSIDAD CATÓLICA DE SANTA MARÍA	PLANO DE COMPONENTE: BRAZO	NOMBRE		FECHA	CÓDIGO DE DIBUJO	REVISION
			DISEÑADO POR:	Rodrigo Alexis Diaz Meza	MAR - 2019		
	DIBUJADO POR:		Rodrigo Alexis Diaz Meza	MAR - 2019	ESCALA: 1:1	HOJA	3 DE 9
	REVISIÓN:						
							A4




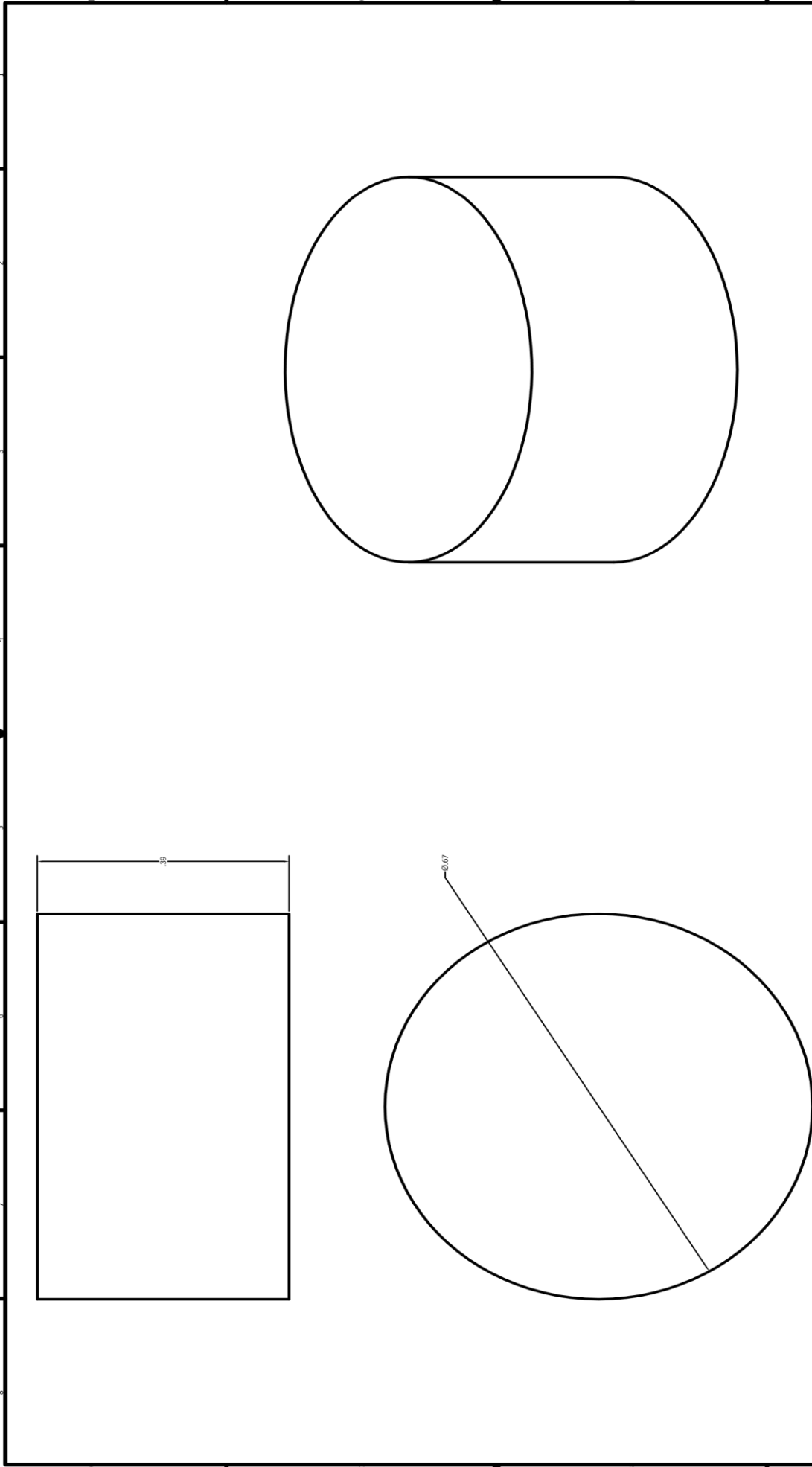
	UNIVERSIDAD CATÓLICA DE SANTA MARÍA	PLANO DE COMPONENTE:	NOMBRE		FECHA	CÓDIGO DE DIBUJO	REVISIÓN
	Ingenierías Físicas y Formales	ROSCA	DISEÑADO POR:	Rodrigo Alexis Díaz Meza	MAR - 2019	USV-004RADM	1.0
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica		DIBUJADO POR:	Rodrigo Alexis Díaz Meza	MAR - 2019	ESCALA: 1:2	A4
			REVISIÓN:			HOJA 4 DE 9	




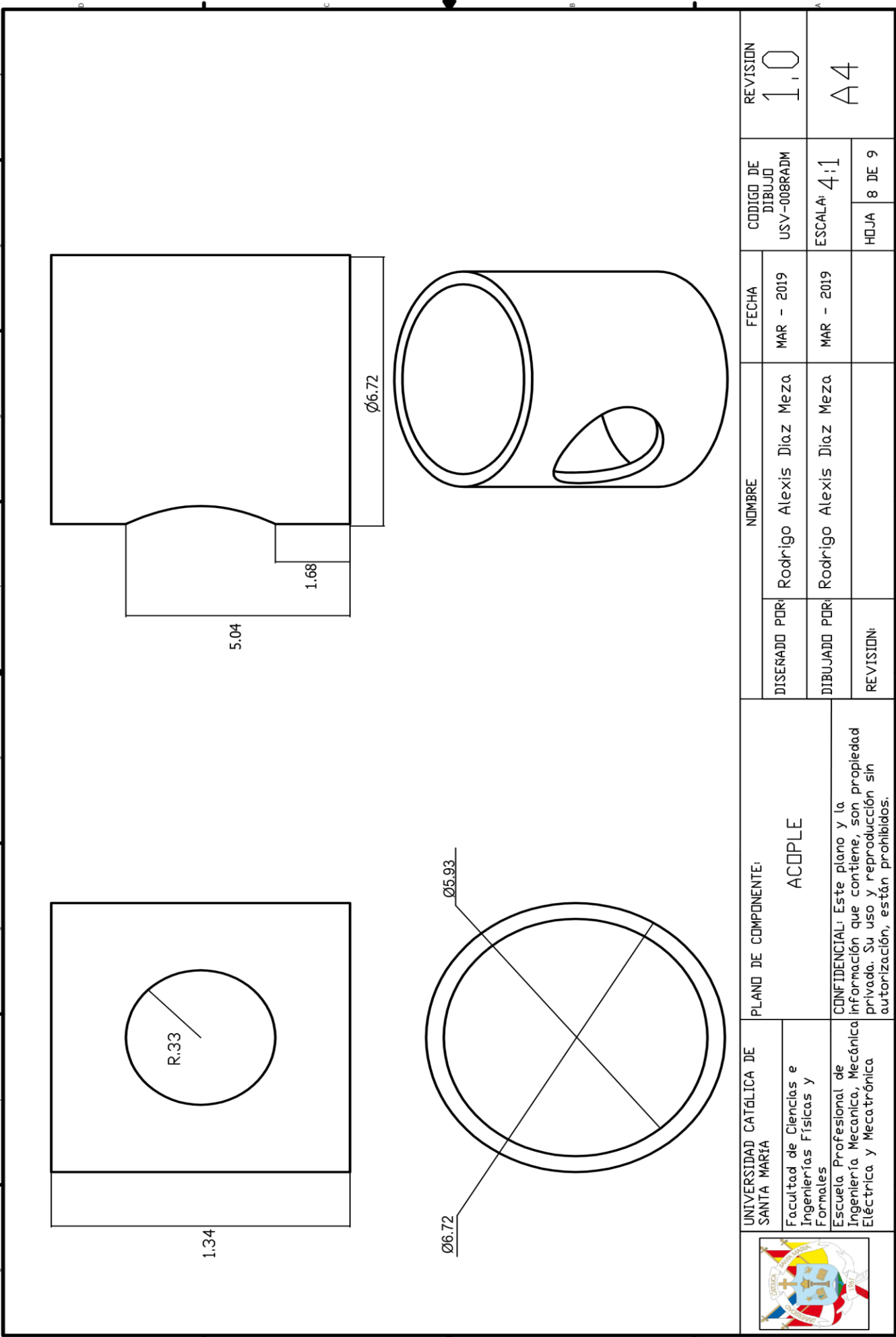
	UNIVERSIDAD CATÓLICA DE SANTA MARÍA	PLANO DE COMPONENTE:	NOMBRE		FECHA	CÓDIGO DE DIBUJO	REVISIÓN
	Facultad de Ciencias e Ingenierías Físicas y Formales	TAPA DELANTERA	DISEÑADO POR:	Rodrigo Alexis Díaz Meza	MAR - 2019	USV-005RADM	1.0
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos.	DIBUJADO POR:	Rodrigo Alexis Díaz Meza	MAR - 2019	ESCALA: 1,65:1	A4
			REVISIÓN:			HOJA 5 DE 9	

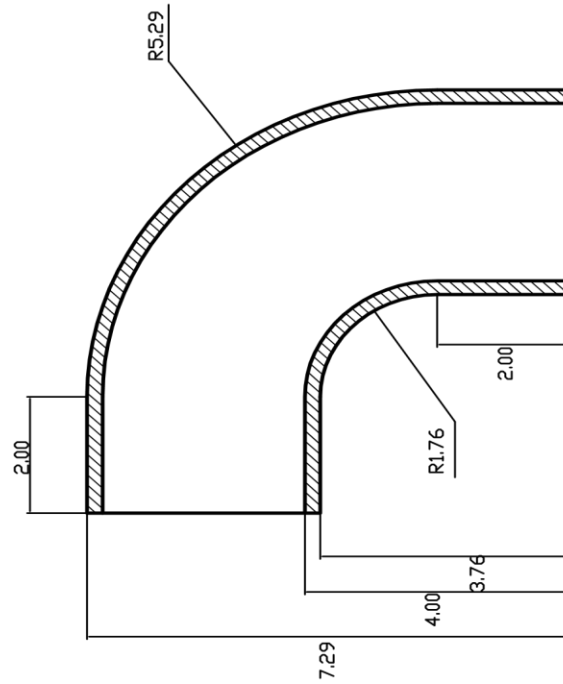
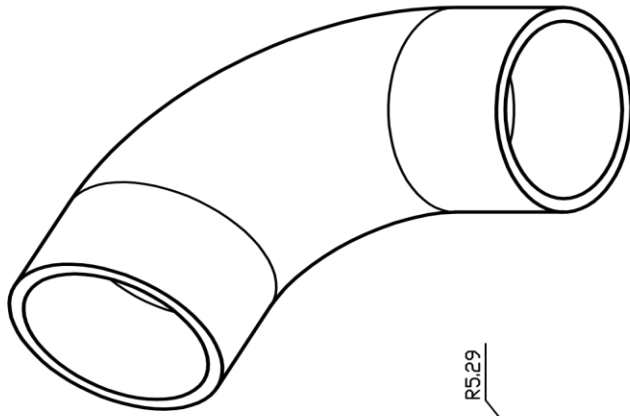


	UNIVERSIDAD CATÓLICA DE SANTA MARÍA	PLANO DE COMPONENTE:		NOMBRE		FECHA	CÓDIGO DE DIBUJO	REVISIÓN			
		TAPA TRASERA				MAR - 2019			USV-006RADM	1,0	
	Facultad de Ciencias e Ingenierías Físicas y Formales			DISEÑADO POR:	Rodrigo Alexis Díaz Meza	DIBUJADO POR:	Rodrigo Alexis Díaz Meza	MAR - 2019	ESCALA: 1,5:1	HOJA 6 DE 9	A4
				Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	DISEÑADO POR:						
	CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos.										

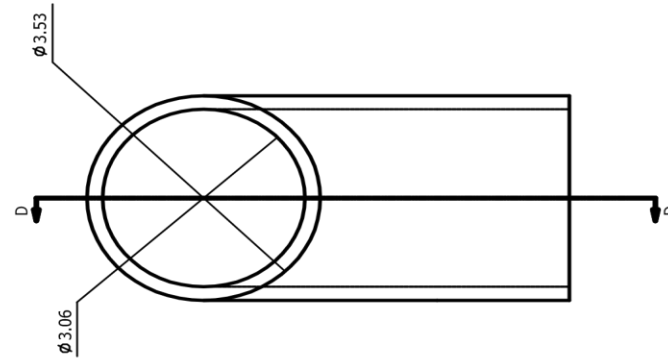



	UNIVERSIDAD CATÓLICA DE SANTA MARÍA		PLANO DE COMPONENTE:		NOMBRE		FECHA	CÓDIGO DE DIBUJO	REVISION
	Facultad de Ciencias e Ingenierías Físicas y Formales		TAPÓN		Rodrigo Alexis Diaz Meza		MAR - 2019		
	Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica		CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos.		Rodrigo Alexis Diaz Meza		MAR - 2019		
					REVISIÓN:				
						ESCALA: 13:1		HOJA	7 DE 9
								A4	





SECTION D-D
SCALE 3:1



	UNIVERSIDAD CATÓLICA DE SANTA MARÍA	PLANO DE COMPONENTE:		CÓDIGO		CONFIDENCIAL: Este plano y la información que contiene, son propiedad privada. Su uso y reproducción sin autorización, están prohibidos.	DISEÑADO POR:		RODRIGO ALEXIS DIAZ MEZA	MAR - 2019	CÓDIGO DE DIBUJO USV-009RADM	REVISION 1.0						
							REVISIÓN:											
			ESCALA: 3:1										HOJA 9 DE 9					
			A4															